

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΤΕ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αριθμός 1496

ΤΙΤΛΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

**Ανάπτυξη ενός Μονοφασικού Αναλυτή Ενέργειας με
το ATmega 328 με δυνατότητα επικοινωνίας με το
Labview μέσω Ethernet.**

**Development of a Single Phase Energy Analyzer
with Atmega 328 which is able to Communicate with
LabVIEW via Ethernet**

ΣΠΟΥΔΑΣΤΗΣ:

ARDENIS FEJZAJ (ΑΡΝΤΕΝΙΣ ΦΕΙΖΑΙ)

ΕΙΣΗΓΗΤΗΣ:

ΑΘΑΝΑΣΙΟΣ ΚΑΛΑΝΤΖΟΠΟΥΛΟΣ

ΠΑΤΡΑ 2015

Πρόλογος

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη ενός μονοφασικού αναλυτή ηλεκτρικής ενέργειας ο οποίος μπορεί να συνδεθεί με το LabVIEW μέσω Ethernet. Οι υπολογισμοί των ηλεκτρικών μεγεθών πραγματοποιούνται από τον μικροελεγκτή ATmega 328 ο οποίος είναι τοποθετημένος στην πλατφόρμα Arduino UNO. Το πρόγραμμα που φορτώθηκε στον μικροελεγκτή αναπτύχθηκε μέσω του περιβάλλοντος προγραμματισμού Arduino IDE. Ενώ για την απομακρυσμένη εμφάνιση των υπολογιζόμενων μεγεθών αναπτύχθηκε ένα γραφικό περιβάλλον με το LabVIEW της National Instruments (NI).

Αρχικά στο πρώτο κεφάλαιο της παρούσας πτυχιακής εργασίας, γίνεται μία εισαγωγή για της ανάγκες της μέτρησης, καταγραφής και ελέγχου της κατανάλωσης ηλεκτρικής ενέργειας. Οι παραπάνω ανάγκες προκάλεσαν το ενδιαφέρον για την ανάπτυξη του έξυπνου αναλυτή ηλεκτρικής ενέργειας που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας. Στο πρώτο κεφάλαιο προσδιορίζονται οι στόχοι και οι προδιαγραφές του αναλυτή ενέργειας. Στο δεύτερο κεφάλαιο παρουσιάζεται αναλυτικά το υλικό που χρησιμοποιήθηκε και αναλύονται τα χαρακτηριστικά και οι δυνατότητες του μικροελεγκτή ATmega 328. Στο τρίτο κεφαλαίο γίνεται μια σύντομη περιγραφή και παρουσίαση των λογισμικών πακέτων που χρησιμοποιήθηκαν τόσο για την ανάπτυξη του κώδικα του μικροεπεξεργαστή όσο και για την ανάπτυξη του γραφικού περιβάλλοντος. Το πρόγραμμα που φορτώθηκε στον μικροελεγκτή και η υλοποίηση του γραφικού περιβάλλοντος που αναπτύχθηκε στο LabVIEW παρουσιάζονται στο τέταρτο κεφαλαίο της παρούσας πτυχιακής εργασίας. Τέλος, στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα που προέκυψαν κατά την λειτουργία του αναλυτή. Επίσης στο συγκεκριμένο κεφάλαιο αναφέρονται μια σειρά από συμπεράσματα που προέκυψαν κατά την υλοποίηση του συγκεκριμένου αναλυτή ενέργειας καθώς και ορισμένες προτάσεις τόσο για την βελτίωση της ακρίβειας των μετρήσεων όσο και για πιθανές μελλοντικές επεκτάσεις του.

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Αθανάσιο Καλαντζόπουλο, για την υποστήριξη, την καθοδήγηση και τις πολύτιμες συμβουλές που μου προσέφερε κατά την διεκπεραίωση της πτυχιακής εργασίας μου. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή Πέτρο Βλαχόπουλο για την καθοδήγηση και τις πολύτιμες συμβουλές του καθώς και όλους τους καθηγητές του τμήματος Ηλεκτρολογίας για τις γνώσεις που μου μετάδωσαν όλα αυτά τα χρόνια. Τέλος, ευχαριστώ τους γονείς μου, την αδερφή μου, την κοπέλα μου και τους φίλους μου που με στήριζαν με κάθε τρόπο σε όλη την διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο τον σχεδιασμό και την ανάπτυξη ενός μονοφασικού αναλυτή ηλεκτρικής ενέργειας ο οποίος ελέγχεται μέσω Ethernet και βασίζεται στην αναπτυξιακή πλατφόρμα Arduino UNO που διαθέτει τον μικροελεγκτή Atmega 328 της Atmel. Με σκοπό τον απομακρυσμένο έλεγχο της παραπάνω εφαρμογής αναπτύχθηκε ένα γραφικό περιβάλλον αλληλεπίδρασης με τον χρήστη (Graphical User Interface – GUI) αξιοποιώντας το λογισμικό πακέτο LabVIEW της National Instruments (NI).

Ο συγκεκριμένος αναλυτής ενέργειας έχει την δυνατότητα να συλλέγει σε πραγματικό χρόνο (real-time) δείγματα τόσο για την τάση δικτύου όσο και για το ρεύμα που διαρρέει το επιλεγμένο ηλεκτρικό φορτίο. Για την συλλογή των δειγμάτων της τάσης δικτύου αξιοποιείται μια κυκλωματική διάταξη που αναλαμβάνει την μετατροπή της σε σήμα σύμφωνα με τις προδιαγραφές του επιλεγμένου μικροελεγκτή. Ενώ για την συλλογή των δειγμάτων του ρεύματος φορτίου αξιοποιείται ένας αισθητήρας ρεύματος που βασίζεται σε στοιχείο hall. Αξιοποιώντας τις παραπάνω μετρήσεις ο συγκεκριμένος αναλυτής ενέργειας υπολογίζει την μέγιστη τιμή, την ελάχιστη τιμή, την ενεργό τιμή και την μέση τιμή τόσο της τάσης δικτύου όσο και του ρεύματος φορτίου. Επίσης είναι σε θέση να υπολογίζει την συχνότητα της τάσης του δικτύου, την φαινόμενη ισχύ, την πραγματική ισχύ, την άεργο ισχύ και τον συντελεστή ισχύος του επιλεγμένου φορτίου. Ο χρήστης έχει την δυνατότητα να παρατηρεί τις παραπάνω μετρήσεις τόσο σε τοπικό επίπεδο μέσω μιας οθόνης υγρών κρυστάλλων (Liquid Crystal Display – LCD) όσο και απομακρυσμένα αξιοποιώντας την εφαρμογή που αναπτύχθηκε με το LabVIEW η οποία επικοινωνεί με τον αναλυτή ενέργειας μέσω Ethernet.

Πινάκας περιεχομένων

Πρόλογος	ii
Περίληψη	iii
Πίνακας περιεχομένων.....	iv
Κεφάλαιο 1. Εισαγωγή	1
1.1 Η αναγκαιότητα της μέτρησης και καταγραφής της ηλεκτρικής ενέργειας.....	1
1.2 Έξυπνοι μετρητές ηλεκτρικής ενέργειας.....	1
1.3 Στόχος της πτυχιακής εργασίας.....	2
Κεφάλαιο 2 Το υλικό του συστήματος.....	5
2.1 Εισαγωγή.....	5
2.2 Η πλατφόρμα Arduino Uno Rev3	5
2.2.1 Ιστορική Αναδρομή.....	5
2.2.2 Τι είναι το Arduino.....	6
2.2.3 Τα χαρακτηριστικά της πλατφόρμας Arduino Uno Rev 3	6
2.3 Ο μικροελεγκτής ATmega328.....	8
2.3.1 Η CPU του μικροελεγκτή ATmega328.....	8
2.3.2 Οι μνήμες του ATmega328.....	10
2.3.3 Ακροδέκτες του ATmega328.....	10
2.3.4 Σειριακή επικοινωνία	10
2.3.5 Ρολόι χρονισμού	12
2.3.6 Ο μετατροπέας αναλογικού σήματος σε ψηφιακό του ATmega328 ..	13
2.4 Arduino Ethernet Shield	19
2.5 Η οθόνη LCD	20
2.6 Ο Αισθητήρας ρεύματος ACS712.....	21
2.7 Διάταξη προσαρμογής της τάσης του δικτύου	22
Κεφάλαιο 3 Το λογισμικό που χρησιμοποιήθηκε.....	25
3.1 Εισαγωγή.....	25
3.2 Το περιβάλλον προγραμματισμού της πλατφόρμας Arduino	25
3.2.1 Οι συντομεύσεις και τα μενού του Arduino IDE	26
3.2.2 Το Sketchbook	27
3.2.3 Οι βιβλιοθήκες	28
3.2.4 Serial Monitor	28
3.2.5 Η δομή του κώδικα	29
3.2.6 Παράδειγμα ενός ολοκληρωμένου προγράμματος	31
3.3 Εισαγωγή στο πρόγραμμα LabVIEW	32
3.3.1 Το περιβάλλον του LabVIEW 2011.....	32
3.3.2 Οι τύποι δεδομένων.....	35
3.3.3 Οι συναρτήσεις του LabVIEW	36
3.3.4 Πρωτόκολλα επικοινωνίας.....	42
Κεφάλαιο 4 Υλοποίηση του συστήματος.....	45
4.1 Εισαγωγή.....	45
4.2 Το πρόγραμμα του ATmega 328.....	45

4.3	To LabVIEW interface	59
4.3.1	To Front Panel της εφαρμογής.....	60
4.3.2	To Block Diagram της εφαρμογής.....	60
Κεφάλαιο 5 Αποτελέσματα – Συμπεράσματα		67
5.1	Εισαγωγή.....	67
5.2	Τοπική λειτουργία	68
5.3	Απομακρυσμένη λειτουργία	69
5.4	Συμπεράσματα – Προτάσεις.....	73
Βιβλιογραφία		75

Κεφάλαιο 1

Εισαγωγή

1.1 Η αναγκαιότητα της μέτρησης και καταγραφής της ηλεκτρικής ενέργειας

Είναι γεγονός ότι οι σύγχρονες ανάγκες του ανθρώπου για την κατανάλωση ενέργειας συνεχώς αυξάνονται. Παρόλα αυτά οι πηγές ενέργειας, οι οποίες είναι διαθέσιμες, λιγοστεύουν σε επικίνδυνο βαθμό. Εκτός από την προφανή λύση εύρεσης νέων μορφών και πηγών ενέργειας, προέκυψε και η ανάγκη παρακολούθησης της κατανάλωσης ηλεκτρικής ενέργειας σε οικιακές ηλεκτρικές εγκαταστάσεις. Η καταγραφή, η μέτρηση και ο έλεγχος των απαιτούμενων μεγεθών πραγματοποιείται με τη χρήση ειδικά σχεδιασμένων συσκευών.

Ένας από τους σημαντικότερους λόγους που καθιστά επιτακτική την αναλυτική μέτρηση των μεγεθών αυτών είναι τα οφέλη που μπορεί να προκύψουν για το περιβάλλον. Η καταγραφή των μετρήσεων και η ανάλυση τους μπορεί να οδηγήσει στην ανάπτυξη διαδικασιών που θα αποφέρουν αφενός εξοικονόμηση ενέργειας και αφετέρου θα συμβάλλουν στην μείωση της εκπομπής ρύπων.

Η εξοικονόμηση και ορθολογική χρήση της ηλεκτρικής ενέργειας και κατά επέκταση των φυσικών πόρων δεν έχει μόνο περιβαλλοντικά οφέλη αλλά και προσωπικά. Πιο συγκεκριμένα, ο εκάστοτε καταναλωτής, μέσω της παρακολούθησης της ατομικής του κατανάλωσης έχει τη δυνατότητα να περιορίσει το αντίστοιχο κόστος, χρησιμοποιώντας τις ενεργοβόρες συσκευές σε συγκεκριμένες χρονικές περιόδους όπου υπάρχει μικρότερη ζήτηση ηλεκτρικής ενέργειας και συνεπώς φθηνότερο τιμολόγιο. Το γεγονός αυτό μπορεί να οδηγήσει τόσο στην βελτίωση της καταναλωτικής συμπεριφοράς όσο και στην ανάπτυξη περιβαλλοντικής συνείδησης.

1.2 Έξυπνοι μετρητές ηλεκτρικής ενέργειας

Η αναγκαιότητα για την παρακολούθηση και την καταγραφή κατανάλωσης ηλεκτρικής ενέργειας οδήγησε στην σχεδίαση και στην κατασκευή μετρητικών συσκευών, οι οποίες έχουν την δυνατότητα να μετρούν τα βασικά ηλεκτρικά μεγέθη. Οι συσκευές αυτές ονομάζονται μετρητές ηλεκτρικής ενέργειας.

Οι απαιτήσεις για ηλεκτρική ενέργεια αυξάνονται συνεχώς με αποτέλεσμα να δημιουργούνται ανάγκες για έλεγχο και διαχείριση. Οι συσκευές οι οποίες έχουν σχεδιαστεί για την ικανοποίηση των παραπάνω αναγκών αποκαλούνται έξυπνοι μετρητές ηλεκτρικής ενέργειας (smart energy meters).

Οι έξυπνοι μετρητές σχεδιάζονται με τέτοιο τρόπο ώστε να μπορούν είτε να αντικαταστήσουν τους ήδη υπάρχοντες μετρητές ηλεκτρικής ενέργειας είτε να χρησιμοποιηθούν συμπληρωματικά. Οι έξυπνες αυτές συσκευές μέτρησης μετρούν κάθε χρονική στιγμή τα βασικά ηλεκτρικά μεγέθη όπως την τάση του δικτύου ηλεκτρικής ενέργειας, το ρεύμα του φορτίου την ισχύ που καταναλώνεται, το συντελεστή ισχύος κλπ. Η επίβλεψη των παραπάνω μεγεθών βοηθάει τους χρήστες να κατανοήσουν τις συνέπειες το να ανάβεις ή να σβήνεις διαφορετικές ηλεκτρικές συσκευές. Επίσης ο καταναλωτής θα μπορεί να διαπιστώνει την κατανάλωση της κάθε μιας συσκευής χωριστά, καθώς ο μετρητής θα δείχνει την αύξηση φορτίου σε περίπτωση που για παράδειγμα ανοίξει ο θερμοσίφωνας ή η κουζίνα. Συνήθως οι Έξυπνοι Μετρητές έχουν επίσης ενσωματωμένη μία οθόνη η οποία μπορεί να τοποθετηθεί σε κάποιο εμφανές σημείο, έτσι ώστε η πρόσβαση του χρήστη στις

πληροφορίες να είναι εύκολη, ενώ ο ίδιος ο μετρητής θα μπορεί να βρίσκεται σε ένα οποιοδήποτε απομακρυσμένο σημείο. Οι νέοι μετρητές θα εγκαθίστανται εντός του σπιτιού ή της πολυκατοικίας και όχι εξωτερικά, καθώς δεν θα χρειάζεται να είναι προσβάσιμοι από τα συνεργεία μέτρησης καθώς θα υπάρχει δυνατότητα τηλεμέτρησης. Δηλαδή αντί να έρχεται ένα συνεργείο και να κάνει τις καταμετρήσεις, ο μετρητής θα αποστέλλει τα στοιχεία της κατανάλωσης απευθείας, ασύρματα ή ενσύρματα σε έναν κεντρικό Ηλεκτρονικό Υπολογιστή (H/Y), όπου και θα γίνεται η επεξεργασία για βέλτιστη διαχείριση του δικτύου. Επιπλέον με τους έξυπνους μετρητές θα δίνεται η δυνατότητα πιο ευέλικτων τιμολογίων ανάλογα με την ώρα ή τη μέρα της εβδομάδας. Δηλαδή αυτό που σήμερα ισχύει και εφαρμόζεται με το λεγόμενο "νυχτερινό τιμολόγιο" θα μπορεί να επεκταθεί και οι χρεώσεις να είναι κλιμακωτές ανάλογα με τις ανάγκες τόσο του καταναλωτή όσο και του εκάστοτε παρόχου.

Στο εμπόριο κυκλοφορούν αρκετά μοντέλα έξυπνων μετρητών ενέργειας. Ορισμένοι από τα αντιπροσωπευτικότερα μοντέλα έξυπνων μετρητών ενέργειας είναι τα παρακάτω:

- FLUKE 1750 και FLUKE 1760
- ACE 5000, ACE 6000, ACE 7000 ή ACTARIS SL7000
- Gran-Electro SS-101 και Gran-Electro SS-301
- CTC 5602 και CTC 5605
- Voltech - PM 3000
- AMPROBE DM-III
- Trinity Oracle Portable Power Analysis
- G4500 BLACKBOX Portable Power Quality Analyzer

Ένας χαρακτηριστικός έξυπνος μετρητής ηλεκτρικής ενέργειας είναι ο FLUKE 1760 Three-Phase Power Quality Recorder. Η συσκευή αυτή αποτελεί έναν τριφασικό μετρητή ποιότητας ισχύος η οποία έχει την ικανότητα μέτρησης τάσεων, ρεύματος, ενέργειας και ισχύος καθώς και ανάλυσης των αντίστοιχων αρμονικών. Διαθέτει 8 εισόδους (κανάλια) για μέτρηση τάσεων και ρευμάτων μαζί με τους αντίστοιχους ακροδέκτες με συχνότητα δειγματοληψία που φθάνει έως 10,24 KHz. Για την επικοινωνία και την μεταφορά δεδομένων μεταξύ της μετρητικής συσκευής και οποιασδήποτε άλλης συσκευής αποθήκευσης δεδομένων, όπως ο H/Y, η συγκεκριμένη μετρητική συσκευή διαθέτει θύρες Ethernet και RS-232. Επίσης συνοδεύεται από κατάλληλο λογισμικό για την ανάλυση και την μελέτη των μετρήσεων. Ένα επίσης πολύ σημαντικό χαρακτηριστικό του παραπάνω μετρητή ηλεκτρικής ενέργειας είναι η διαθέσιμη μνήμη των 2 GB, η οποία αξιοποιείται για την αποθήκευση των μετρήσεων. Εκτός όμως από το FLUKE 1760 υπάρχουν και άλλα μοντέλα έξυπνων μετρητών ηλεκτρικής ενέργειας τα οποία εκτός του ότι διαθέτουν εξίσου υψηλή συχνότητα δειγματοληψίας, διαθέτουν και πρόσθετα τεχνικά χαρακτηριστικά όπως η δυνατότητα ασύρματης σύνδεσης, ενσωματωμένο Web Server, FTP Server κλπ.

1.3 Στόχος της πτυχιακής εργασίας

Ο στόχος της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη ενός έξυπνου μονοφασικού αναλυτή ηλεκτρικής ενέργειας ο οποίος θα έχει την δυνατότητα να συνδεθεί με το LabVIEW (Laboratory Virtual Instrument Engineering Workbench) μέσω Ethernet. Ο στόχος του συστήματος αυτού είναι να μετρά και να υπολογίζει τα ηλεκτρικά μεγέθη μιας ηλεκτρικής εγκατάστασης αξιοποιώντας τον μικροελεγκτή ATmega328 ο οποίος είναι τοποθετημένος στην πλατφόρμα Arduino Uno. Στην συνέχεια τα αποτελέσματα των παραπάνω υπολογισμών θα σταλούν στο περιβάλλον που έχει αναπτυχθεί στο LabVIEW μέσω του Ethernet Shield. Στο περιβάλλον του LabVIEW ο χρήστης έχει την δυνατότητα απομακρυσμένης παρακολούθησης και ελέγχου της κατανάλωσης ηλεκτρικής ενέργειας και

την παρατήρηση των μεταβολών των ηλεκτρικών μεγεθών. Μια επιπλέον δυνατότητα του συγκεκριμένου συστήματος είναι η αυτόματη αποσύνδεση ορισμένων μη κρίσιμων φορτίων όταν η συνολική κατανάλωση υπερβαίνει τα όρια που έχει ορίσει ο χρήστης. Επίσης ορισμένα ζητήματα τα οποία οδήγησαν στην μελέτη και κατασκευή μιας τέτοιας μετρητικής διάταξης είναι το χαμηλό κόστος σε σύγκριση με τους υπάρχοντες αναλυτές ενέργειας, το μικρό μέγεθος, η χρηστικότητα, και η ελάχιστη κατανάλωση ενέργειας της συσκευής.

Τα βασικά μέρη του συγκεκριμένου μονοφασικού αναλυτή ηλεκτρικής ενέργειας, τα οποία θα αναλυθούν στα επόμενα κεφάλαια, είναι ο μικροελεγκτής AT mega328 ο οποίος βρίσκεται στην πλατφόρμα arduino, το Ethernet ShieldR3, η LCD (Liquid Crystal Display) οθόνη QC2004A τεχνολογίας υγρών κρυστάλλων, ο αισθητήρας ρεύματος ACS712, και η διάταξη προσαρμογής της τάσης του δικτύου.

Κεφάλαιο 2

Το υλικό του συστήματος

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνεται μια αναλυτική περιγραφή του υλικού που χρησιμοποιείται για την ανάπτυξη του έξυπνου μονοφασικού αναλυτή ηλεκτρικής ενέργειας. Για κάθε διάταξη που περιλαμβάνει ο έξυπνος μονοφασικός αναλυτής ηλεκτρικής ενέργειας πραγματοποιείται μια σύντομη αναφορά τόσο σε θεωρητικό επίπεδο όσο και στον τρόπο λειτουργίας της. Για την υλοποίηση του συγκεκριμένου συστήματος αξιοποιήθηκε η πλατφόρμα Arduino Uno Rev3, το Arduino Ethernet Shield, μια LCD οθόνη χαρακτήρων, ο αισθητήρας ρεύματος ACS712 που βασίζεται στο στοιχείο Hall και μια διάταξη για την προσαρμογή της τάσης του δικτύου στα 5V ώστε να είναι δυνατή η μέτρηση της τάσης από τον μικροελεγκτή που διαθέτει η πλατφόρμα Arduino Uno. Επίσης πραγματοποιείται μια σύντομη αναφορά στις δυνατότητες του μικροελεγκτή ATmega 328 που διαθέτει η συγκεκριμένη πλατφόρμα εστιάζοντας κυρίως στις περιφερειακές μονάδες που αξιοποιήθηκαν στα πλαίσια της παρούσας πτυχιακής εργασίας.

2.2 Η πλατφόρμα Arduino Uno Rev3

Στο σχήμα 2.1 παρουσιάζεται η πλατφόρμα Arduino Uno Rev3 που χρησιμοποιείται στην παρούσα πτυχιακή εργασία και είναι ένα από τα πλέον δημοφιλή μέλη της οικογένειας Arduino. Οι δυνατότητες της παρουσιάζουν μεγάλο ενδιαφέρον και αναλύονται στην συνέχεια αυτής της παραγράφου.



Σχήμα 2.1: Οι δύο όψεις της πλατφόρμας Arduino Uno

2.2.1 Ιστορική Αναδρομή

Το Arduino ξεκίνησε σε ένα μικρό εργοστάσιο στην πόλη της Ιβρέα (Interaction Design Institute Ivrea γνωστό και ως Interaction Ivrea, IDII ή Ivrea), η οποία είναι μια κομόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας. Στην ίδια περιοχή είχε ως έδρα επίσης και η εταιρία υπολογιστών Olivetti. Στόχος ήταν να υλοποιηθεί μία πλατφόρμα για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από τα άλλα συστήματα που ήταν διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν την συγκεκριμένη πλατφόρμα Arduino από τον Arduin της Ιβρέα, βασιλιά της Ιταλίας.

2.2.2 Τί είναι το Arduino

Το Arduino θα μπορούσε να θεωρηθεί ως μια open-source αναπτυξιακή πλατφόρμα η οποία ενσωματώνει ένα μικροελεγκτή και έχει την δυνατότητα να συνδέεται με ένα Η/Υ. Μέσω του Η/Υ ο χρήστης έχει την δυνατότητα να προγραμματίσει τον αντίστοιχο μικροελεγκτή αξιοποιώντας το διαθέσιμο αναπτυξιακό περιβάλλον του Arduino. Η συγκεκριμένη πλατφόρμα μπορεί να αξιοποιηθεί για την ανάπτυξη διαδραστικών εφαρμογών που αξιοποιούν πληθώρα αισθητηρίων και έχουν την δυνατότητα να ελέγχουν εξωτερικές διατάξεις και συσκευές. Οι παραπάνω εφαρμογές μπορούν να είναι αυτόνομες ή να ελέγχονται μέσω κάποιου προγράμματος στον Η/Υ του χρήστη.

2.2.3 Τα χαρακτηριστικά της πλατφόρμας Arduino Uno Rev 3

Το Arduino Uno R3 είναι μια υπολογιστική πλατφόρμα βασισμένη στο μικροελεγκτή ATmega328 της Atmel. Έχει 14 ψηφιακούς ακροδέκτες εισόδου και εξόδου από τους οποίους οι 6 μπορούν να χρησιμοποιηθούν ως PWM (Pulse Width Modulation) έξοδοι. Διαθέτει επίσης 6 αναλογικές εισόδους οι οποίες χρησιμοποιούνται για την ανάγνωση αναλογικών τιμών από εξωτερικά στοιχεία όπως ποτενσιόμετρα, αναλογικοί αισθητήρες κτλ. Επιπλέον η πλατφόρμα Arduino Uno περιλαμβάνει ένα κεραμικό κρύσταλλο στα 16 MHz, μία θύρα USB (Universal Serial Port), μια υποδοχή για εξωτερική τροφοδοσίας, μια ICSP (In Circuit Serial Programming) σύνδεση και ένα κουμπί reset. Επίσης το Arduino Uno χρησιμοποιεί το μικροελεγκτή ATmega16U2 που αναλαμβάνει τον ρόλο του μετατροπέα από USB σε Serial. Τα γενικά χαρακτηριστικά της πλατφόρμας Arduino Uno παρουσιάζονται στον πίνακα 2.1.

Γενικά χαρακτηριστικά της πλατφόρμας Arduino Uno Rev 3	
Microcontroller	ATmega328
CPU	8- bit AVR
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Πίνακας 2.1 – Γενικά χαρακτηριστικά της πλατφόρμας Arduino Uno Rev3

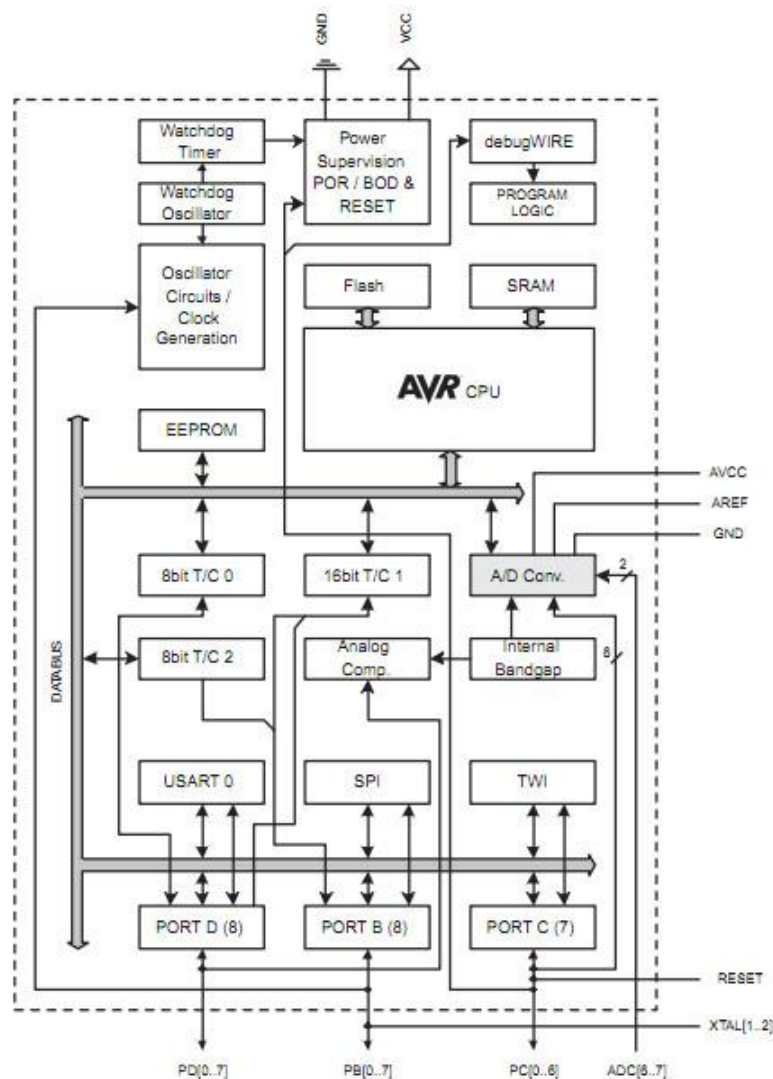
Η υπολογιστική πλατφόρμα Arduino Uno R3 εκτός από τους ακροδέκτες εισόδου – εξόδου έχει επίσης κάποιους ακροδέκτες οι οποίοι έχουν συγκεκριμένες λειτουργίες. Στον πίνακα 2.2 αναλυεται ο σκοπός και οι λειτουργίες κάθε ακροδέκτη της παραπάνω πλατφόρμας.

Ακροδέκτης	Λειτουργία	Περιγραφή
0-13	Ψηφιακοί είσοδοι-έξοδοι	Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος, χρησιμοποιώντας τις συναρτήσεις pinMode (), digitalWrite (), και digitalWrite (). Λειτουργούν στα 5 V με ρεύμα έως 20 mA επίσης διαθέτουν και pull-up αντιστάσεις 20-50kΩ
A0-A5	Αναλογικοί είσοδοι	Διαβάζουν αναλογικά σήματα τάσης από 0-5V επιστρέφοντας μια ακέραιη τιμή από 0 έως 1023 (10-bit ακρίβεια).
3.3V	Τροφοδοσία +3.3V	Η DC τάση +3.3 V προέρχεται από το σταθεροποιητή τάσης της πλατφόρμας Arduino.
5V	Τροφοδοσία +5V	Η DC τάση 5V προέρχεται από το σταθεροποιητή τάσης που διαθέτει η πλατφόρμα Arduino. Επίσης ο ίδιος ακροδέκτης μπορεί να χρησιμοποιηθεί για την απευθείας τροφοδότηση της πλατφόρμας παρακάμπτοντας το σταθεροποιητή τάσης.
GND	Γείωση (0V)	Όλες οι διαφορές δυναμικού της πλατφόρμας είναι σε σχέση με τον ακροδέκτη GND που χρησιμοποιείται ως σημείο αναφοράς.
Vin	Εξωτερική τροφοδοσία	Ο ακροδέκτης Vin χρησιμοποιείται ως τάση τροφοδοσίας για την πλατφόρμα Arduino όταν η τροφοδοσία δεν προέρχεται από την USB θύρα.
IOREF	+5 V	Παρέχει την τάση αναφοράς του μικροελεγκτή. Μια συσκευή που συνδέεται στην πλατφόρμα μπορεί να διαβάσει την τάση IOREF και επιλέξει την κατάλληλη τάση τροφοδοσίας.
0→ RX 1→ TX	Σειριακή Επικοινωνία	Χρησιμοποιούνται για λήψη (RX) και αποστολή (TX) σειριακών δεδομένων. Οι ακροδέκτες αυτοί συνδέονται με τους αντίστοιχους ακροδέκτες του ολοκληρωμένου FTDI USB to-TTL Serial.
2 και 3	Εξωτερικές Διακοπές (Interrupts)	Ενεργοποιούν την αντίστοιχη διακοπή κάθε φορά που ανιχνεύεται παλμός τάσης. Οι ενεργοποίηση των διακοπών πραγματοποιείται με την συνάρτηση attachInterrupt().
3, 5, 6, 9, 10, 11	PWM	Παρέχουν μια PWM έξοδο με Duty Cycle από 0-100% αξιοποιώντας την συνάρτηση analogWrite() η οποία δέχεται ως όρισμα ένα 8-bit ακέραιο αριθμό χωρίς πρόσημο.
10(SS), 11(MOSI), 12(MISO), 13(SCK)	SPI	Αυτοί οι ακροδέκτες επιτρέπουν την SPI επικοινωνία η οποία υποστηρίζεται μέσω της SPI βιβλιοθήκης που διαθέτει το περιβάλλον προγραμματισμού του Arduino.
13	LED	Ο ακροδέκτης 13 είναι συνδεδεμένος με ένα LED που διαθέτει η πλατφόρμα Arduino.
4 (SDA) 5 (SCL).	I2C	Υποστηρίζει το πρωτόκολλο I2C χρησιμοποιώντας βιβλιοθήκες τις Γλώσσας προγραμματισμού Wiring.
AREF	Εξωτερική τάση αναφοράς	Χρησιμοποιείται ως εξωτερική τάση αναφοράς μέσω της συνάρτησης analogReference()
RESET	Reset.	Αν τεθεί σε κατάσταση LOW τότε επανενεκινεί τον Μικροελεγκτή.

Πίνακας 2.2 – Τα χαρακτηριστικά των ακροδεκτών της πλατφόρμας Arduino Uno.

2.3 Ο μικροελεγκτής ATmega328

Ο ATmega328 μικροελεγκτής ανήκει στην κατηγορία των "Low Power Atmel AVR 8-bit Microcontrollers" και είναι ο μικροελεγκτής που αξιοποιείται για την υλοποίηση του συγκεκριμένου αναλυτή ηλεκτρικής ενέργειας μέσω της πλατφόρμας Arduino Uno. Είναι των 8 bit και βασίζεται στην προχωρημένη RISC (Reduced Instruction Set Computing) αρχιτεκτονική. Όπως παρουσιάζεται στο σχήμα 2.2, ο πυρήνας του ATmega328 αποτελείται από την CPU (Central Processing Unit), την μνήμη SRAM (Static Random Access Memory), την μνήμη EEPROM (Electrically Erasable Programmable Read Only Memory), την μνήμη Flash και ένα σύνολο περιφερειακών μονάδων. Ποιο συγκεκριμένα για τα περιφερικά του ATmega328 θα αναφερθούμε στην συνέχεια του κεφαλαίου.



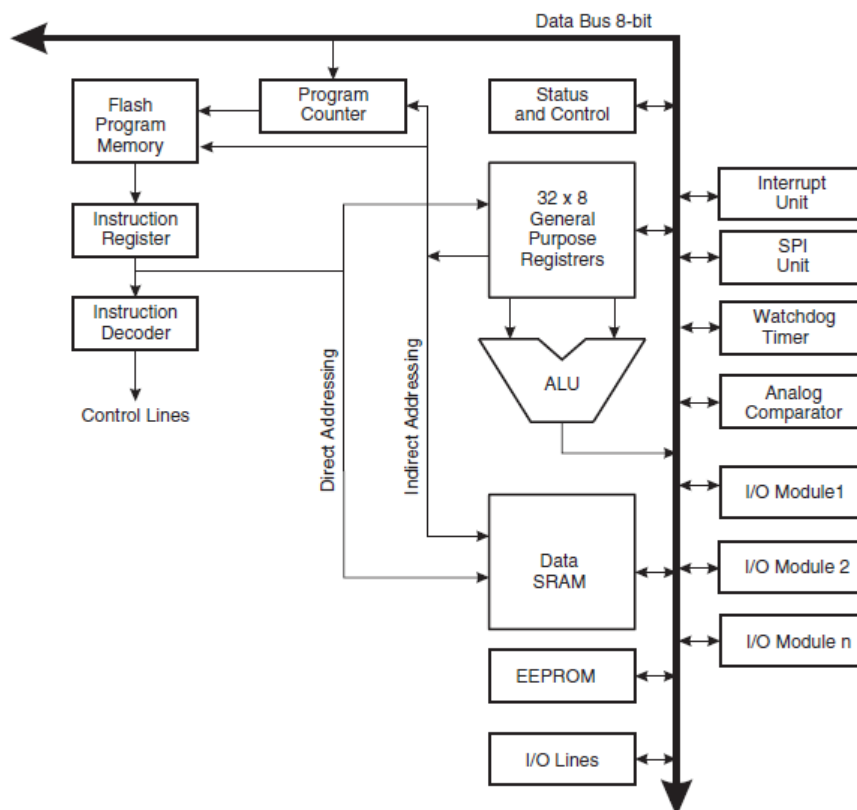
Σχήμα 2.2: Μπλοκ διάγραμμα ενός AVR μικροελεγκτή

2.3.1 Η CPU του μικροελεγκτή ATmega328

Σε αυτή την ενότητα γίνεται μια σύντομη περιγραφή της CPU του ATmega328 η οποία βασίζεται στην AVR αρχιτεκτονική (σχήμα 2.3) που είναι μια τροποποιημένη Harvard αρχιτεκτονική.

Η βασική λειτουργία της CPU του ATmega328 είναι η διασφάλιση της σωστής εκτέλεσης του προγράμματος. Η CPU έχει επίσης την δυνατότητα να προσπελαίνει τις διαθέσιμες μνήμες, να εκτελεί υπολογισμούς, να ελέγχει τις περιφερειακές μονάδες, και να διαχειρίζεται τις διακοπές (interrupts). Στον πυρήνα της CPU υπάρχει μια ALU (Arithmetic Logic Unit - Αριθμητική Λογική Μονάδα) η οποία υποστηρίζει αριθμητικές και λογικές πράξεις μεταξύ των καταχωρητών (registers) ή μεταξύ μιας σταθεράς και ενός καταχωρητή. Διαθέτει επίσης ένα καταχωρητή κατάστασης (Status Register) όπου τα bit του αλλάζουν κατάσταση ανάλογα με το αποτέλεσμα της ALU. Για παράδειγμα το bit 1 γίνεται HIGH όταν το αποτέλεσμα της ALU είναι 0. Επίσης στην CPU του ATmega328 υπάρχει ένα αρχείο καταχωρητών γενικού σκοπού (General Purpose Register File) προκειμένου να επιτευχθεί η απαιτούμενη απόδοση και ευελιξία της CPU. Επιπλέον υπάρχουν 32 καταχωρητές, από τους οποίους οι 6 μπορούν να χρησιμοποιηθούν ως τρεις 16-bit καταχωρητές έμμεσης διευθυνσιοδότησης που υποδεικνύουν την περιοχή δεδομένων.

Η AVR αρχιτεκτονική υποστηρίζει πολλαπλές πηγές διακοπής, πιο συγκεκριμένα ο ATmega328 υποστηρίζει 26 διαφορετικές διακοπές. Η διαχείριση των υποστηριζόμενων διακοπών πραγματοποιείται από μια ευέλικτη μονάδα διακοπών η οποία διαθέτει του δικούς της καταχωρητές ελέγχου και ενεργοποιείται από το Global Interrupt Enable bit του καταχωρητή κατάστασης. Σε κάθε διακοπή αντιστοιχεί ένα διάνυσμα διακοπής (interrupt vector) στον πίνακα διανυσμάτων διακοπής (Interrupt Vector Table). Η θέση του διανύσματος διακοπής στον παραπάνω πίνακα καθορίζει την προτεραιότητα της αντίστοιχής διακοπής. Κατά την εξυπηρέτηση μιας διακοπής η CPU διακόπτει την εκτέλεση του προγράμματος και ξεκινά την εκτέλεση της συνάρτησης εξυπηρέτησης της διακοπής. Όταν ολοκληρωθεί η εκτέλεση της συνάρτησης εξυπηρέτησης διακοπής, η CPU συνεχίζει την εκτέλεση του προγράμματος από το σημείο που είχε σταματήσει.



Σχήμα 2.3 Διάγραμμα της αρχιτεκτονικής AVR

2.3.2 Οι μνήμες του ATmega328

Ο μικροελεγκτής ATmega328 διαθέτει 3 είδη μνημών τα οποία έχουν διαφορετικά χαρακτηριστικά και χρησιμοποιούνται για διαφορετικές λειτουργίες.

Για την αποθήκευση του προγράμματος ο ATmega328 διαθέτει μια μνήμη Flash των 32 KB. Στην περίπτωση της πλατφόρμας Arduino Uno τα 0.5KB χρησιμοποιούνται για την αποθήκευση του bootloader. Ο συγκεκριμένος bootloader είναι απαραίτητος για την φόρτωση των προγραμμάτων που θα εκτελέσει ο μικροελεγκτής μέσω της θύρας USB. Επομένως για την αποθήκευση των προγραμμάτων του χρήστη αξιοποιούνται τα υπόλοιπα 31.5KB της μνήμης Flash. Το ιδιαίτερο χαρακτηριστικό της μνήμης Flash είναι ότι δε χάνει τα δεδομένα της μετά από απώλεια της τάσης τροφοδοσίας ή επανεκκίνηση.

Ο ATmega328 αξιοποιεί την μνήμη SRAM των 2KB για την αποθήκευση των δεδομένων του προγράμματος. Συνεπώς τα δεδομένα των μεταβλητών ή των πινάκων του προγράμματος αποθηκεύονται στην μνήμη SRAM. Η συγκεκριμένη μνήμη χάνει τα δεδομένα της όταν διακοπεί η τάση τροφοδοσίας του μικροελεγκτή ή όταν πατηθεί το κουμπί reset.

Τέλος ο ATmega328 διαθέτει 1KB μνήμη EEPROM, η οποία χρησιμοποιείται για εγγραφή ή ανάγνωση δεδομένων. Σε αντίθεση με την μνήμη SRAM, δε χάνει τα δεδομένα της μετά από απώλεια της τάσης τροφοδοσίας ή επανεκκίνηση.

2.3.3 Ακροδέκτες του ATmega328

Στο σχήμα 2.4 φαίνεται η αντιστοιχία των ακροδεκτών του μικροελεγκτή ATmega328 και της πλατφόρμας Arduino Uno. Ο ATmega328 διαθέτει συνολικά 28 ακροδέκτες ενώ η πλατφόρμα Arduino Uno διαθέτει 32. Η πλατφόρμα Arduino Uno περιλαμβάνει όλους τους ακροδέκτες του παραπάνω μικροελεγκτή, μερικοί από αυτούς είναι συνδεδεμένοι με τις εσωτερικές μονάδες της πλατφόρμας, ενώ οι υπόλοιποι είναι διαθέσιμοι μέσω της πλατφόρμας.

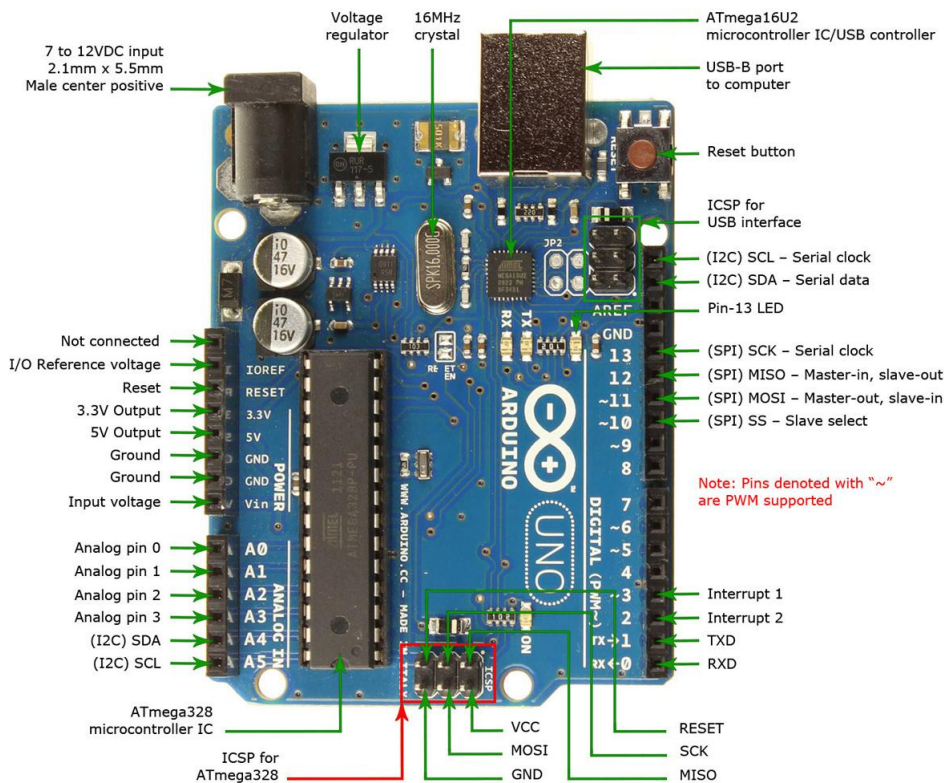
2.3.4 Σειριακή επικοινωνία

Ο ATmega328 υποστηρίζει τρία είδη σειριακής επικοινωνίας αξιοποιώντας τις μονάδες USART (Universal Synchronous and Asynchronous Receiver and Transmitter), SPI (Serial Peripheral Interface) και την TWI (Two Wire serial Interface). Κάθε μια από τις παραπάνω μονάδες αξιοποιείται σε διαφορετικές εφαρμογές ανάλογα με τις εκάστοτε απαιτήσεις της εφαρμογής αλλά και τα χαρακτηριστικά της συγκεκριμένης μονάδας.

Τα χαρακτηριστικά της μονάδας USART είναι τα παρακάτω:

- Πλήρης αμφίδρομη λειτουργία (ανεξάρτητοι σειριακοί καταχωρητές λήψης και μετάδοσης)
- Ασύγχρονη ή σύγχρονη λειτουργία
- Αφέντη (master) ή σκλάβου (slave) σύγχρονη λειτουργία
- Υψηλής ανάλυσης γεννήτριας ρυθμού μετάδοσης (baud rate generator)
- Η παραγωγή μονής ή ζυγής ισοτιμίας αλλά και ο έλεγχος ισοτιμίας υποστηρίζονται από το υλικό
- Ανίχνευση υπέρβασης ορίου μεταφοράς δεδομένων (Data OverRun Detection)
- Ανίχνευση σφαλμάτων διαμόρφωσης μηνυμάτων (Framing Error Detection)
- Φιλτράρισμα θορύβου που περιλαμβάνει ανίχνευση λανθασμένου bit εκκίνησης και ψηφιακό βαθυπερατό φίλτρο

Arduino function		Arduino function	
reset	(PCINT14/RESET) PC6	1	28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25 PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24 PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23 PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22 GND GND
GND	GND	8	21 AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20 AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17 PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)



Σχήμα 2.4 – Ο χάρτης των ακροδεκτών του Arduino Uno και του μικροελεγκτή ATmega328

- Υποστηρίζει τρεις διακριτές διακοπές την TX Complete, την TX Data Register Empty και την RX Complete
- Λειτουργία επικοινωνίας πολλαπλών επεξεργαστών (Multi-processor Communication Mode)
- Λειτουργία ασύγχρονης επικοινωνίας διπλής ταχύτητας (Double Speed Asynchronous Communication Mode)

Η μονάδα USART του ATmega328 χρησιμοποιείται μέσω της πλατφόρμας Arduino για την επικοινωνία με τον Η/Υ. Επίσης χρησιμοποιείται για την φόρτωση του προγράμματος στο Arduino αξιοποιώντας τον bootloader. Η μονάδα USART μπορεί να αξιοποιηθεί και για την επικοινωνία με συσκευές που υποστηρίζουν σειριακή επικοινωνία.

Μια επιπλέον μονάδα επικοινωνίας που διαθέτει ο ATmega328 είναι η SPI (Serial Peripheral Interface), η οποία αξιοποιείται για μεταφορά δεδομένων με υψηλή ταχύτητα μεταξύ περιφερικών μονάδων και άλλων μικροελεγκτών. Στην συνέχεια αναφέρονται μερικά από τα βασικά χαρακτηριστικά της μονάδας SPI του ATmega328.

- Τριών καλωδίων πλήρης αμφίδρομη σύγχρονη μεταφορά δεδομένων
- Λειτουργία αφέντη ή σκλάβου
- Μεταφορά δεδομένων με πρώτο τον λιγότερο σημαντικό Byte ή το περισσότερο σημαντικό Byte
- Διαθέτει 7 προγραμματιζόμενους ρυθμούς μετάδοσης
- Ενεργοποιεί μια διακοπή κατά την ολοκλήρωση της μετάδοσης
- Δυνατότητα ενεργοποίησης από την κατάσταση αναμονής
- Διπλή ταχύτητα κατά την SPI λειτουργία ως αφέντης

Η 3 σειριακή μονάδα επικοινωνίας που υποστηρίζει ο ATmega328 είναι η TWI η οποία παρουσιάζει τα παρακάτω χαρακτηριστικά:

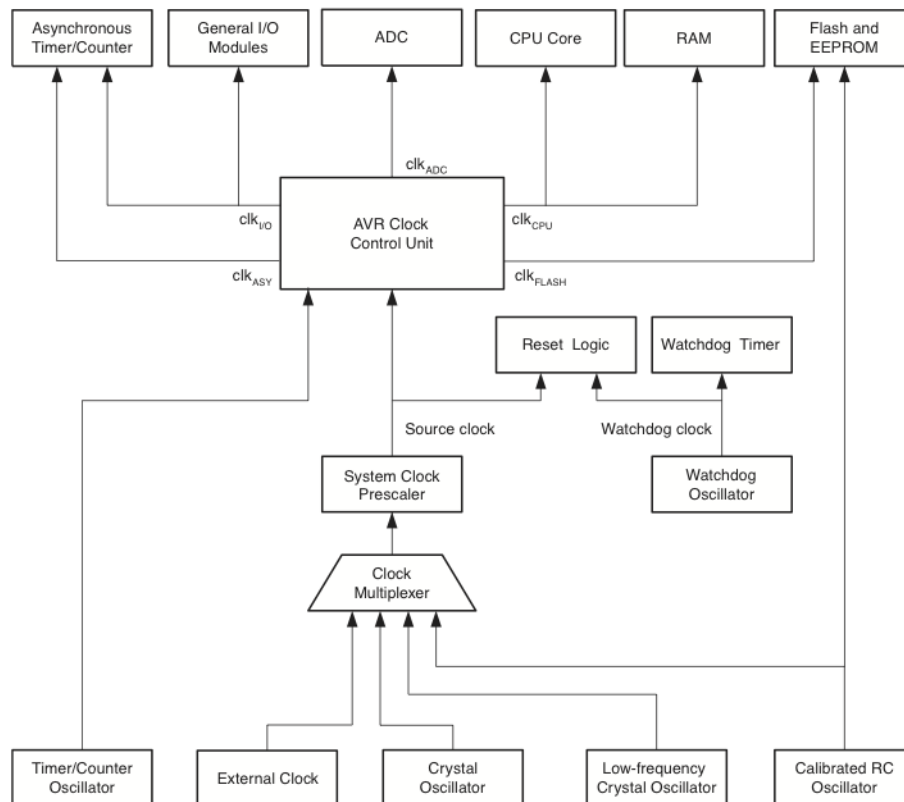
- Απλή, ισχυρή και ευέλικτη επικοινωνία μέσω μόνο 2 καλωδίων.
- Υποστηρίζει λειτουργία αφέντη ή σκλάβου.
- Η συσκευή έχει τη δυνατότητα λειτουργίας ως πομπός ή δέκτης.
- Διαθέτει χώρο διεύθυνσης των 7 bit επιτρέποντας έως 128 διεύθυνσης για σκλάβους.
- Υποστηρίζει διαχείριση πολλαπλών αφεντών.
- Μέγιστη ταχύτητα μεταφοράς δεδομένων 400kHz.
- Διαθέτει κύκλωμα μείωσης αιχμής θορύβου στους δίαυλους μετάδοσης.
- Πλήρης προγραμματιζόμενη διεύθυνση σκλάβου με υποστήριξη γενικής πρόκλησης.
- Διαθέτη αναγνώριση διευθύνσεων με σκοπό την επαναλειτουργία του AVR όταν βρίσκετε σε αναμονή
- Συμβατό με το πρωτόκολλο I2C (Inter-Integrated Circuit) της εταιρίας Philips

Η σειριακή μονάδα επικοινωνίας TWI είναι προσαρμοσμένη για τυπικές εφαρμογές του μικροελεγκτή. Το TWI επιτρέπει την διασύνδεση έως και 128 διαφορετικές συσκευές χρησιμοποιώντας μόνο δύο καλώδια επικοινωνίας. Το ένα καλώδιο χρησιμοποιείται για το ρολόι (SCL) και το δεύτερο για τα δεδομένα (SDA), όπου για κάθε μία από τις γραμμές απαιτητέ μια pull-up εξωτερική αντίσταση. Όλες οι συσκευές που συνδέονται στους διαύλους επικοινωνίας έχουν ξεχωριστές διευθύνσεις, καθώς και διαθέτουν μηχανισμούς συμβατότητας με το πρωτόκολλο TWI.

2.3.5 Ρολόι χρονισμού

Σε πολλά ψηφιακά κυκλώματα η σειρά με την οποία πραγματοποιούνται τα συμβάντα είναι κρίσιμη. Μερικές φορές κάποιο συμβάν πρέπει να προηγείται κάποιου άλλου, ενώ μερικές φορές δύο συμβάντα πρέπει να πραγματοποιούνται ταυτόχρονα. Για να μπορούν οι σχεδιαστές να πετύχουν τις απαιτούμενες χρονικές σχέσεις, πολλά ψηφιακά κυκλώματα χρησιμοποιούν ρολόγια για την επίτευξη του χρονισμού. Υπάρχουν τέσσερις τύποι ρολογιών που χρησιμοποιούνται στους μικροελεγκτές, οι κρύσταλλοι (crystals), οι κεραμικοί ταλαντωτές (ceramic resonators), οι RC (Resistor, Capacitor) ταλαντωτές και ταλαντωτές πυριτίου (silicon oscillators).

Στο σχήμα 2.5 παρουσιάζονται τα κύρια συστήματα του ρολογιού του μικροελεγκτή ATmega328 και η διανομή των σημάτων ρολογιού στα επιμέρους περιφερειακά ψηφιακά κυκλώματα και μονάδες. Για εξοικονόμηση ενέργειας, τα σήματα ρολογιού στις επιμέρους περιφερειακές μονάδες που δεν χρησιμοποιούνται απενεργοποιούνται σύμφωνα με την επιλεγμένη κατάσταση αναμονής (sleep mode).



Σχήμα 2.5 – Διανομή των σημάτων ρολογιού

Το κυρίως ρολόι χρονισμού του ATmega328 έχει συχνότητα που κυμαίνεται από μερικά MHz μέχρι και 20 MHz που είναι η ονομαστική συχνότητα του συγκεκριμένου μικροελεγκτή. Στην συνέχεια περιγράφονται τα βασικά σήματα χρονισμού του μικροελεγκτή ATmega328.

Το ρολόι της CPU συνδέεται σε μέρη του συστήματος τα οποία σχετίζονται με την λειτουργία του πυρήνα του μικροελεγκτή όπως είναι οι καταχωρητές γενικού σκοπού, ο καταχωρητής κατάστασης ή μνήμη δεδομένων που περιέχει τον δείκτη σωρού (Stack Pointer) κλπ. Σε περίπτωση που το ρολόι χρονισμού της CPU διακοπεί τότε η CPU διακόπτεται την εκτέλεση διαφόρων λειτουργιών και υπολογισμών.

Το ρολόι εισόδων –εξόδων (I/O clock) χρησιμοποιείται κύριως από τις μονάδες εισόδων –εξόδων (I/O modules), όπως τους μετρητές και την μονάδα SPI και την μονάδα USART. Το ρολόι εισόδων– εξόδων χρησιμοποιείται επίσης και από τη μονάδα εξωτερικών διακοπών.

Ο ADC (Analog to Digital Converter) του ATmega328 είναι συνδεδεμένος με ανεξάρτητο ρολόι χρονισμού. Το γεγονός αυτό επιτρέπει το σταμάτημα του ρολογιού της CPU και του ρολογιού εισόδων –εξόδων με σκοπό τη μείωση του θορύβου που παράγεται από τα ψηφιακά κυκλώματα ώστε να είναι πιο ακριβή τα αποτελέσματα των μετατροπών του ADC.

2.3.6 Ο μετατροπέας αναλογικού σήματος σε ψηφιακό του ATmega328

Καθώς ο στόχος της πτυχιακής είναι να επιτευχθεί η κατασκευή μίας ψηφιακής μετρητικής διάταξης αναλογικών σημάτων πραγματικού χρόνου, αυτόματα προκύπτει η ανάγκη για την αξιοποίηση του ADC που διαθέτει ο ATmega328. Ο συγκεκριμένος ADC έχει τα παρακάτω χαρακτηριστικά:

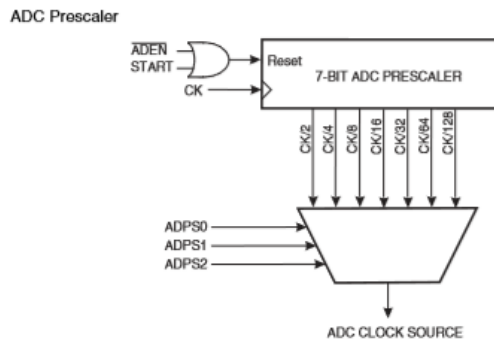
- 10-bit ανάλυση.
- Ο χρόνος μετατροπής κυμαίνεται από 13 έως 260 μs

- Μέγιστη ταχύτητα μετατροπής 76,9 kSPS (ή 15 kSPS στην μέγιστη ανάλυση)
- Το εύρος των μετρήσιμων αναλογικών τιμών κυμαίνεται από 0-V_{CC}
- Διαθέτει 6 πολυπλεγμένα κανάλια εισόδου
- Διαθέτει ένα κανάλι εισόδου για αισθητήρα θερμοκρασίας
- Δυνατότητα αξιοποίησης της 1,1 V εσωτερικής τάσης αναφοράς
- Δυνατότητα επιλογής λειτουργίας (Free Running ή Single Conversion)
- Ενεργοποιεί μια διακοπή όταν ολοκληρώνεται μια μετατροπή

Συμφώνα με τα παραπάνω χαρακτηριστικά ο ADC του ATmega328 είναι των 10 bit δηλαδή οι αναλογικές τιμές που διαβάζει μετατρέπονται σε ακέραιους αριθμούς με τιμή από 0 έως 1023. Η τιμή 0 αντιστοιχεί σε τάση 0V ενώ η τιμή 1023 αντιστοιχεί σε τάση 5V αν η τάση αναφοράς του ADC είναι 5V. Έχει μικρό σχετικά χρόνο μετατροπής με αποτέλεσμα να μπορεί να υποστηρίξει σχετικά υψηλές συχνότητες δειγματοληψίας. Επίσης ο συγκεκριμένος ADC έχει δυνατότητα ενεργοποίησης μιας διακοπής κάθε φορά που ολοκληρώνεται μια μετατροπή. Επιπλέον υποστηρίζει δύο τρόπους λειτουργίας τον Free Running και τον Single Conversion. Σύμφωνα με τον Free Running τρόπο λειτουργίας μια νέα μετατροπή ξεκινά αμέσως μετά την προηγούμενη λαμβάνοντας υπόψη την επιθυμητή συχνότητα δειγματοληψίας. Αντίθετα κατά τον τρόπο λειτουργίας Single Conversion μια νέα μετατροπή θα ξεκινήσει έπειτα από εντολή του προγράμματος που εκτελεί ο μικροελεγκτής. Για να είναι εφικτή η δειγματοληψία του ρεύματος και της τάσης από τον συγκεκριμένο αναλυτή ενέργειας σε πραγματικό χρόνο επιλέχτηκε ο Free Running τρόπος λειτουργίας και η αξιοποίηση της διακοπής που προκαλείται κάθε φορά που ο ADC ολοκληρώνει μια μετατροπή.

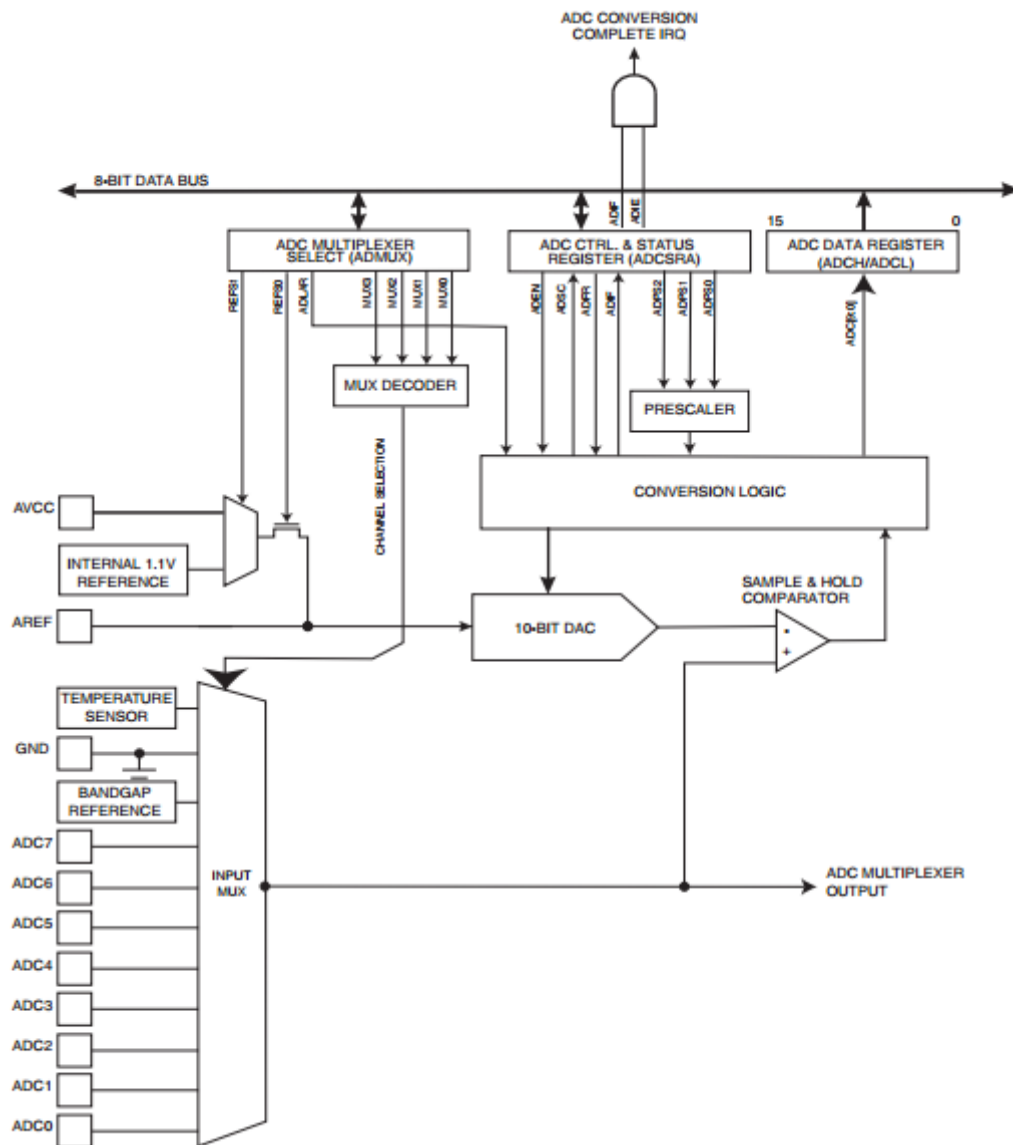
Συχνότητα δειγματοληψίας

Το ρολόι του μικροελεγκτή ATmega 328 που στην πλατφόρμα Arduino είναι 16MHz υποδιαιρείται ανάλογα με την τιμή του prescaler όπως παρουσιάζεται στο σχήμα 2.6. Πιο συγκεκριμένα αν η τιμή του prescaler είναι 7 το ρολόι του μικροελεγκτή διαιρείται με την τιμή 128. Επομένως η συχνότητα του ρολογιού του ADC είναι $16 \text{ MHz}/128=125\text{kHz}$. Για την ολοκλήρωση κάθε μετατροπής απαιτούνται 13 κύκλοι ρολογιού με αποτέλεσμα η συχνότητα δειγματοληψίας του ADC στο συγκεκριμένο παράδειγμα να είναι $125\text{kHz}/13 = 9,6 \text{ kHz}$. Οι απαιτήσεις του συγκεκριμένου αναλυτή ενέργειας είναι να διαβάσει δυο αναλογικούς εισόδους μια για την τάση και μια για το ρεύμα. Συνεπώς, η συχνότητα δειγματοληψίας του αναλυτή ενέργειας είναι η μισή σε σχέση με την συχνότητα δειγματοληψίας του ADC. Το γεγονός αυτό οφείλεται στο ότι οι 6 διαθέσιμες αναλογικές εισοδοί του ADC που διαθέτει ο ATmega328 χρησιμοποιούνται μέσω ενός πολυπλέκτη όπως παρουσιάζεται στο σχήμα 2.7.



Σχήμα 2.6 – ADC prescaler

Η συχνότητα του δικτύου ηλεκτρικής ενέργειας στην Ευρώπη είναι 50Hz που αντιστοιχεί σε περίοδο 20ms. Συνεπώς ο αναλυτής ενέργειας που έχει συχνότητα δειγματοληψίας $F_s = 9,6\text{kHz}/2 = 4,8\text{kHz}$ θα συλλέξει $\text{Samples} = 20\text{ms} * 4,8 \text{ kHz} = 96$ δείγματα ανά περίοδο τάσης ή ρεύματος.



Σχήμα 2.7 -Schematic diagram του ADC του AT mega328.

Η διακοπή του ADC

Ο ADC κατά την ολοκλήρωση μιας μετατροπής ενεργοποιεί μια σημαία (flag) η οποία βρίσκεται στην θέση μνήμης 0x002A του μικροελεγκτή ATmega328. Όταν η σημαία αυτή ενεργοποιείται τότε η CPU διακόπτει την εκτέλεση του κυρίως προγράμματος και ξεκάνει την εκτέλεση της συνάρτησης εξυπηρέτησης διακοπής του ADC (ISR(ADC_vect)). Όταν η συνάρτηση εξυπηρέτησης της διακοπής ολοκληρωθεί η CPU συνεχίζει την εκτέλεση του προγράμματος από το σημείο που είχε σταματήσει.

Το αποτέλεσμα του ADC

Το αποτέλεσμα της μετατροπής το οποίο πραγματοποιεί ο ADC του ATmega328 είναι ένας ψηφιακός αριθμός των 10 bit και για την αποθήκευση του δεν επαρκεί μόνο ένας 8bit καταχωρητής. Έτσι για την αποθήκευση του συγκεκριμένου αποτελέσματος χρησιμοποιούνται οι καταχωρητές ADCL και ADCH. Η σειρά αποθήκευσης του αποτελέσματος στους παραπάνω καταχωρητές εξαρτάται από την τιμή που έχει το bit ADLAR το οποίο ανήκει στον καταχωρητή ADMUX. Οι δυο δυνατοί τρόποι αποθήκευσης του αποτελέσματος παρουσιάζονται αναλυτικά στους πίνακες 2.9 και 2.10.

Το αποτέλεσμα της μετατροπής του ADC σχετίζεται με την επιλεγμένη τάση αναφοράς του ATmega328 και υπολογίζεται σύμφωνα με την παρακάτω σχέση:

$$\text{ADC result} = V_{\text{IN}} \cdot 1024 / V_{\text{REF}}$$

Όπου το ADC result είναι το αποτέλεσμα της μετατροπής, το V_{IN} είναι η τιμή της αναλογικής τάσης εισόδου και το V_{REF} είναι η επιλεγμένη τάση αναφοράς. Για τον συγκεκριμένο αναλυτή ενέργειας, ως τάση αναφοράς του ADC χρησιμοποιείται η τάση τροφοδοσίας της πλατφόρμας Arduino Uno, δηλαδή $V_{\text{REF}} = 5\text{V}$. Στην περίπτωση αυτή προκύπτει ότι η διακριτική ικανότητα του μετατροπέα να είναι:

$$5 / 1024 = 4.8828125 \text{ mV}$$

Οι καταχωρητές του ADC

Η επιλογή του τρόπου λειτουργίας και η παραμετροποίηση του ADC που διαθέτει ο ATmega328 πραγματοποιείται μέσω των 8-bit καταχωρητών ADMUX, ADCSRA, ADCSRB, και DIDR0. Οι καταχωρητές αυτοί ανάλογα με το συνδυασμό των bits τους πραγματοποιούν την παραμετροποίηση του ADC.

ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0
0x7C	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Πίνακας 2.3 Ο καταχωρητής ADMUX

Bit 7:6 → REFS1, REFS0: Τα συγκεκριμένα bit χρησιμοποιούνται για την επιλογή της τάσης αναφοράς που συνδέεται στον ADC. οι διαθέσιμες επιλογές παρουσιάζονται στον πίνακα 2.4.

REFS1	REFS0	Επιλογή τάσης αναφοράς (V_{REF})
0	0	AREF (Η εσωτερική τάση αναφοράς είναι απενεργοποιημένη.)
0	1	AV_{cc} με σύνδεση εξωτερικού πυκνωτή στον ακροδέκτη AREF.
1	0	Reserved
1	1	Εσωτερική 1.1V με σύνδεση εξωτερικού πυκνωτή στον ακροδέκτη AREF.

Πίνακας 2.4 – Επιλογές της τάσης αναφοράς (V_{REF}) του ADC

- Bit 5 → ADLAR: Το bit αυτό καθορίζει το τρόπο αποθήκευσης του αποτελέσματος της μετατροπής του ADC στους καταχωρητές ADCL και ADCH. Οι δυο δυνατοί τρόποι αποθήκευσης παρουσιάζονται στους πίνακες 2.9 και 2.10.

- Bit 4→Reserved.
- Bit 0:3→MUX[3:0]: Ο συνδυασμός των bit αυτών καθορίζει την αναλογική είσοδο από τον οποίο θα πραγματοποιηθεί η επόμενη μετατροπή.

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0
0x7C	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Πίνακας 2.5 - Ο καταχωρητής ADCSRA

- Bit 7 → ADEN: Όταν το bit αυτό λάβει την τιμή 1 τότε πραγματοποιείται η ενεργοποίηση του ADC αλλιώς ο ADC είναι απενεργοποιημένος. Επιπλέον απενεργοποιώντας το συγκεκριμένο bit κατά τη διάρκεια της μετατροπής οδηγεί στην διακοπή της μετατροπής.
- Bit 6→ADSC: Αρχίζει την μετατροπή από αναλογικό σε ψηφιακό.
- Bit 5→ADATE: Όταν το bit 5 του καταχωρητή ADATE πάρει την λογική τιμή 1 τότε η συνεχόμενη λειτουργία του ADC έχει ενεργοποιηθεί, δηλαδή ο ADC θα πραγματοποιεί μια μετατροπή σε κάθε ανερχόμενο μέτωπο του ρολογιού του.
- Bit 4→ADIF: Το συγκεκριμένο bit ενεργοποιείται όταν ο ADC έχει ολοκληρώσει την μετατροπή και οι καταχωριτές δεδομένων έχουν ανανεώσει τις τιμές τους
- Bit 3→ADIE : Όταν το bit ADIE και το bit SREG γίνουν 1 τότε ενεργοποιείται η διακοπή που αντιστοιχεί στην ολοκλήρωση της μετατροπής.
- Bit 2:0→ADPS1: Τα bit αυτά καθορίζουν τον παράγοντα διαιρέτη (prescaler) που θα χρησιμοποιηθεί, στον πίνακα 2.6 φαίνονται οι διαθέσιμες επιλογές ανάλογα με το συνδυασμό των bit **ADPS2**, **ADPS1** και **ADPS0**.

ADPS2	ADPS1	ADPS0	Παράγοντας διαιρέτη (prescaler)
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Πίνακας 2.6 – Επιλογές του παράγοντα διαιρέτη.

ADCSR_B – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0
0x7C	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Πίνακας 2.7 - Ο καταχωρητής ADCSR_B

- Bit 7,3:5 → Τα bit αυτά είναι διαθέσιμα για μελλοντική χρήση.
- Bit 2:0 → ADTS[2:0]: μέσω των bit 0 έως 2 του συγκεκριμένου καταχωριστή πραγματοποιείται η επιλογή της πηγής ενεργοποίησης του ADC, οι οκτώ δυνατές συνδυασμοί που μπορούν να φτιάξουν τα συγκεκριμένα bit αντιστοιχούν σε όγδο διαφορετικές πηγές ενεργοποίησης του ADC. Η αντιστοιχία του κάθε συνδυασμού με την πηγή ενεργοποίησης παρουσιάζεται στον πίνακα 2.8.

ADTS2	ADTS1	ADTS0	Πηγή ενεργοποίησης του ADC
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Πίνακας 2.8 - Πηγή ενεργοποίησης του ADC

ADCH και ADCL – The ADC Data Register

Όπως αναφέρθηκε και στην προηγούμενη παράγραφο το αποτέλεσμα της μετατροπής του ADC αποθηκεύεται σε δύο καταχωριστές, στον ADCL και στον ADCH. Ανάλογα με την τιμή του bit ADLAR(του καταχωρητή ADMUX) τα δεδομένα αποθηκεύονται είτε με το σημαντικότερο ψηφίο να ξεκινήσει από το ADCL είτε από το ADCH. Ποιο συγκεκριμένα ο ακριβής τρόπος αποθήκευσης του αποτελέσματος παρουσιάζεται στους πίνακες 2.9 και 2.10.

Για **ADLAR** = 0

Bit	15	14	13	12	11	10	9	8
0x79 ADCH	-	-	-	-	-	-	ADC9	ADC8
0x78 ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Bit	7	6	5	4	3	2	1	0
Read/Write ADCH	R	R	R	R	R	R	R	R
Read/Write ADCL	R	R	R	R	R	R	R	R
Initial value ADCL	0	0	0	0	0	0	0	0
Initial value ADCL	0	0	0	0	0	0	0	0

Πίνακας 2.9 - Οι καταχωριστές ADCH και ADCL

Για **ADLAR = 1**

Bit	15	14	13	12	11	10	9	8
0x79 ADCH	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
0x78 ADCL	ADC1	ADC0	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Read/Write ADCH	R	R	R	R	R	R	R	R
Read/Write ADCL	R	R	R	R	R	R	R	R
Initial value ADCL	0	0	0	0	0	0	0	0
Initial value ADCL	0	0	0	0	0	0	0	0

Πίνακας 2.10 - Οι καταχωριτές ADCH και ADCL

DIDR0 – Digital Input Disable Register 0

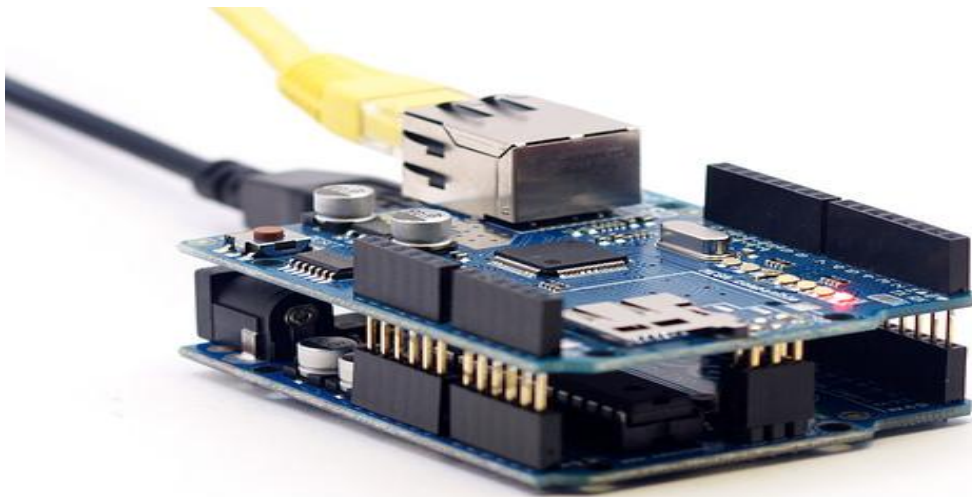
Bit	7	6	5	4	3	2	1	0
0x7E	-	-	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Πίνακας 2.3 Ο καταχωρητής DIDR0

- Bit 7:6→Τα bit αυτά είναι διαθέσιμα για μελλοντική χρήση.
- Bit 5:0→Όταν κάποιο από αυτά τα bit έχει την τιμή '1' η αντίστοιχη αναλογική είσοδος απενεργοποιείται.

2.4 Arduino Ethernet Shield

Το Arduino Ethernet Shield (σχήμα 2.9) είναι μια ηλεκτρονική διάταξη η οποία αναλαμβάνει την μετάδοση των δεδομένων από το Arduino στο διαδίκτυο. Η πλακέτα αυτή επικοινωνεί με την πλατφόρμα Arduino μέσω της θύρας SPI. Ο συγκεκριμένος τρόπος επικοινωνίας αξιοποιεί τους ακροδέκτες 10,11,12,13 της πλατφόρμας Arduino Uno. Η τάση λειτουργίας του Ethernet Shield είναι στα 5V και παρέχεται από την πλατφόρμα Arduino. Το συγκεκριμένο Ethernet Shield διαθέτει τον ελεγκτή Ethernet W5100 που έχει ενσωματωμένη μνήμη 16 kB. Ο συγκεκριμένος Ethernet ελεγκτής της Wiznet παρέχει την στοίβα δικτύου (IP) που είναι ικανή για την υποστήριξη των TCP (Transmission Control Protocol - Πρωτόκολλο Ελέγχου Μεταφοράς) και UDP (User Datagram Protocol or Universal Datagram Protocol) πρωτοκόλλων με ταχύτητα επικοινωνίας 10/100 Mb. Το Ethernet Shield έχει την δυνατότητα να υποστηρίζει μέχρι και 5 ταυτόχρονες συνδέσεις τύπου socket. Τέλος, το Ethernet Shield σε συνδυασμό με την βιβλιοθήκη Ethernet αξιοποιούνται στην συγκεκριμένη πτυχιακή εργασία για την μετάδοση των δεδομένων από την πλατφόρμα Arduino UNO στο περιβάλλον του LabVIEW μέσω διαδικτύου.



Σχήμα 2.9 – Arduino Uno μαζί με το Ethernet Shield

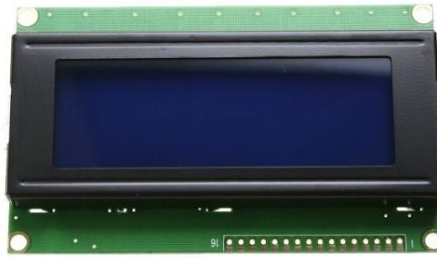
Το Ethernet Shield διαθέτει μια θύρα επικοινωνίας τύπου Rj-45 οπύ συνδέεται το καλώδιο Ethernet και ένα κουμπί για την επανεκκίνηση της. Επίσης διαθέτει μια ενσωματωμένη υποδοχή για κάρτα μνήμης micro SD η οποία μπορεί να χρησιμοποιηθεί για την αποθήκευση αρχείων ή δεδομένων.

Το Ethernet Shield διαθέτει επιπλέον και μια σειρά από LED όπου ο ρόλος τους είναι να παράγουν βοηθητικές ενδείξεις.

- | | | |
|--------------|---|---|
| PWR | → | Δείχνει ότι το Ethernet Shield έχει τροφοδοσία. |
| LINK | → | Δείχνει αν υπάρχει συνδεδεμένο καλώδιο δικτύου και επίσης αναβοσβήνει όταν το Ethernet Shield δέχεται ή στέλνει δεδομένα. |
| FULLD | → | Δείχνει αν η σύνδεση είναι πλήρως αμφίδρομη. |
| 100Mb | → | Δείχνει την ταχύτητα της σύνδεσης στο δίκτυο, αν είναι αναμμένο τότε υποστηρίζεται ταχύτητα 100Mb. |
| RX | → | Αναβοσβήνει όταν το Ethernet Shield λαμβάνει δεδομένα. |
| TX | → | Αναβοσβήνει όταν το Ethernet Shield στέλνει δεδομένα. |
| COLL | → | Αναβοσβήνει όταν εντοπιστεί κάποια διένεξη στο δίκτυο. |

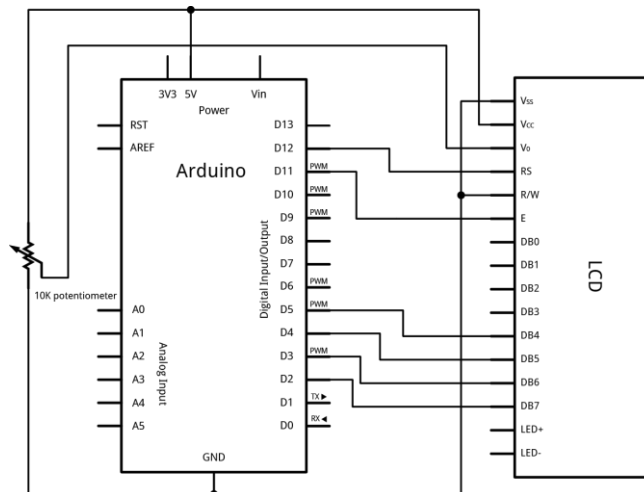
2.5 Η οθόνη LCD

Στην τελική διάταξη έχει προσαρμοστεί η οθόνη τεχνολογίας υγρών κρυστάλλων QC2004A η οποία παρουσιάζεται στο σχήμα 2.10. Το συγκεκριμένο μοντέλο υποστηρίζει τέσσερις γραμμές χαρακτήρων και είκοσι χαρακτήρες ανά γραμμή (4x20) σε μπλε φόντο με λευκούς χαρακτήρες. Για την λειτουργία της απαιτείται στον στο πρόγραμμα που θα φορτωθεί στον Arduino να συμπεριλαμβάνεται η βιβλιοθήκη Liquid Crystal. Η χρήση της οθόνης για τον συγκεκριμένο αναλυτή ενέργειας είναι να εμφανίζει τα αποτελέσματα σε τοπικό επίπεδο δηλαδή στον χώρο που πραγματοποιούνται οι μετρήσεις.



Σχήμα 2.10: Η LCD οθόνη QC2004A

Στο σχήμα 2.11 παρουσιάζεται το σχέδιο για την συνδεσμολογία του ποτενσιόμετρου και των ακροδεκτών της οθόνης με την πλατφόρμα Arduino. Η χρήση του ποτενσιόμετρου είναι για την ρύθμιση της αντίθεσης της οθόνης. Η συνδεσμολογία αυτή απευθύνεται στις περισσότερες LCD οθόνες που είναι συμβατές με τις πλατφόρμες Arduino. Τέλος στον πίνακα 2.11 παρουσιάζονται τα σύμβολα των ακροδεκτών της οθόνης και η αντιστοιχία τους με τους ακροδέκτες της πλατφόρμας Arduino Uno.



Σχήμα 2.11: Η σύνδεση της LCD οθόνης με το Arduino

LCD PIN	Σύμβολο	Arduino pin
LCD PIN 1	Rss	Στη γείωση του arduino
LCD PIN 2	Vdd	Στα 5v του arduino
LCD PIN 3	V0	Μεσαίο pin ποτενσιόμετρου
LCD PIN 4	Rs	Arduino Pin 12
LCD PIN 5	Rw	Στη γείωση του arduino
LCD PIN 6	E	Arduino Pin 11
LCD PIN 7 - 10	-	Δεν χρησιμοποιείτε
LCD PIN 11	D4	Arduino Pin 5
LCD PIN 12	D5	Arduino Pin 4
LCD PIN 13	D6	Arduino Pin 3
LCD PIN 14	D7	Arduino Pin 2
LCD PIN 15	-	-
LCD PIN 16	A	Στα 3,3v του Arduino

Πίνακας 2.11 - Οι ακροδέκτες της οθόνης LCD

2.6 Ο Αισθητήρας ρεύματος ACS712

Για την μέτρηση του ρεύματος χρησιμοποιείται ο αισθητήρας ACS712 (σχήμα 2.12) του οποίου η λειτουργία βασίζεται στο στοιχείο Hall. Το μικρό ποσοστό σφάλματος, ο μικρός

χρόνος απόκρισης, η ελάχιστη αντίσταση που παρουσιάζει ο αισθητήρας και το χαμηλό κόστος του είναι τα βασικά πλεονεκτήματα τα οποία οδήγησαν στην επιλογή του συγκεκριμένου αισθητήρα με σκοπό την ικανοποίηση των αναγκών της παρούσας πτυχιακής εργασίας. Ποιο συγκεκριμένα τα χαρακτηριστικά του αισθητήρα ACS712 αναφέρονται επιγραμματικά παρακάτω:

- Αναλογικό σήμα χαμηλού θορύβου
- 5 μ s χρόνος απόκρισης
- Εύρος ζώνης συχνοτήτων λειτουργίας 80 kHz
- Συνολικό σφάλμα εξόδου 1,5% σε θερμοκρασία 25°C
- Μικρό μέγεθος, χαμηλού προφίλ SOIC8 πακέτο
- 1,2 m Ω εσωτερική αντίσταση αγωγού
- 2.1 kV RMS ελάχιστη τάση απομόνωσης από τους ακροδέκτες 1-4 με 5-8
- Λειτουργεί στα 5.0 V, με ενιαία παροχή
- 66 mV/A ευαισθησία εξόδου (για αισθητήρα των 30 A)
- Η τάση εξόδου είναι ανάλογη με τα AC ή DC ρεύματα
- Εξαιρετικά σταθερή τάση εξόδου
- Σχεδόν μηδενική μαγνητική υστέρηση

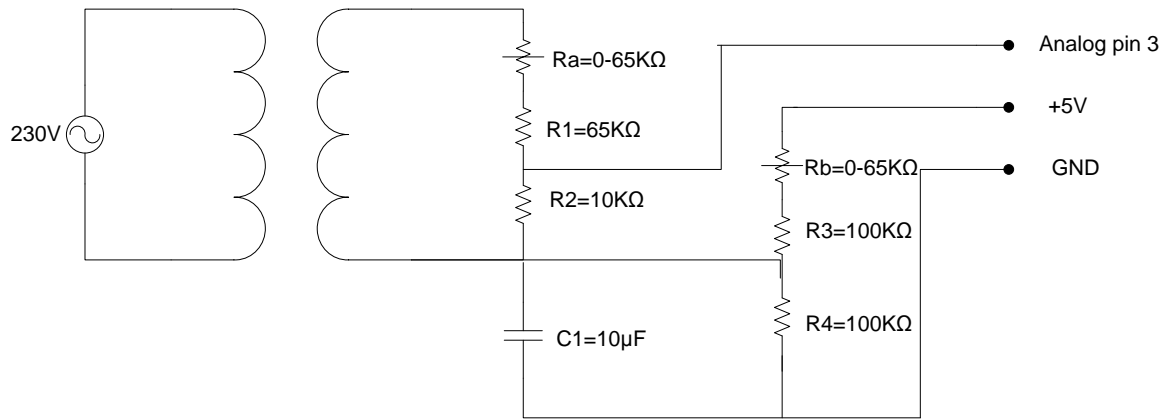


Σχήμα 2.12 – Ο αισθητήρας ρεύματος ACS712, τυπικό διάγραμμα.

Η εσωτερική αντίσταση της αγωγίσιμης διαδρομής του αισθητήρα είναι 1,2 m Ω και έχει ως αποτέλεσμα τον περιορισμό των απωλειών ισχύος. Επιπλέον το πάχος του χάλκινου αγωγού επιτρέπει την λειτουργία του συγκεκριμένου αισθητήρα σε συνθήκες υπερέντασης μέχρι και 5 φορές πάνω από το ονομαστικό ρεύμα. Οι ακροδέκτες της αγωγίσιμης διαδρομής είναι ηλεκτρικά απομονωμένοι από τον αισθητήρα. Το γεγονός αυτό επιτρέπει στον αισθητήρα ACS712 να χρησιμοποιείται σε εφαρμογές που απαιτούν ηλεκτρική απομόνωση.

2.7 Διάταξη προσαρμογής της τάσης του δικτύου

Για να είναι εφικτή η μέτρηση της τάσης του δικτύου από τον μικροελεγκτή ATmega 328 θα πρέπει να γίνει η μετατροπή της στα όρια των αναλογικών σημάτων που μπορεί να μετρήσει ο συγκεκριμένος μικροελεγκτής. Το εύρος τιμών των αναλογικών σημάτων που μπορεί να διαβάσει ο μικροελεγκτής ATmega 328 είναι από 0V έως 5V. Έτσι για τον υποβιβασμό της τάσης του δικτύου χρησιμοποιείται ένας μετασχηματιστής υποβιβασμού με λόγο 230V/12 V. Στην συνέχεια στην έξοδο του μετασχηματιστή συνδέεται ένας διαιρετής τάσης ο οποίος διαιρεί την τάση των 12V σε 3Vpp. Επειδή ο ADC του ATmega 328 δεν έχει την δυνατότητα μέτρησης αρνητικών τιμών προστίθεται στην τάση εξόδου του διαιρέτη τάση μια DC συνιστώσα με τιμή 2,5V. Με τον τρόπο αυτό επιτυγχάνεται η προσαρμογή της τάσης δικτύου στο αποδεκτό εύρος τιμών που μπορούν να μετρηθούν από το συγκεκριμένο ADC. Τέλος στο σχήμα 2.13 παρουσιάζεται το κύκλωμα το οποίο αναλαμβάνει την παραπάνω προσαρμογή.



Σχήμα 2.13 – Το κύκλωμα προσαρμογής της τάσης δικτύου

Κεφάλαιο 3

Το λογισμικό που χρησιμοποιήθηκε

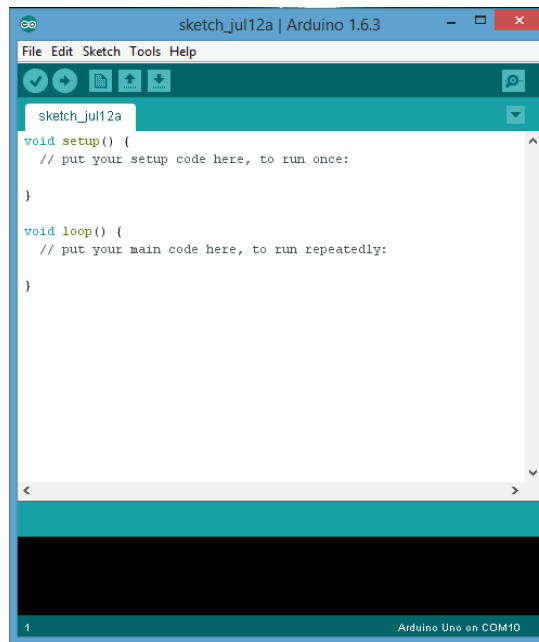
3.1 Εισαγωγή

Στο κεφαλαίο αυτό γίνεται μια σύντομη αναφορά στα πακέτα λογισμικού που χρησιμοποιήθηκαν στα πλαίσια της παρούσας πτυχιακής εργασίας, το Arduino IDE (Integrated Development Environment) και το LabVIEW της NI (National Instruments). Στο περιβάλλον του Arduino IDE αναπτύχθηκε το πρόγραμμα του Arduino που αναλαμβάνει την συνεχή δειγματοληψία, τον υπολογισμό των ηλεκτρικών μεγεθών και την αποστολή τους μέσω Ethernet αξιοποιώντας το Ethernet Shield. Ενώ το λογισμικό πακέτο LabVIEW χρησιμοποιήθηκε για την υλοποίηση του GUI (Graphical User Interface) το οποίο είναι υπεύθυνο για την λήψη των δεδομένων που αποστέλλονται μέσω του Ethernet Shield και την απεικόνιση τους είτε ως μεμονωμένες τιμές είτε ως γραφικές παραστάσεις. Επίσης το GUI παρέχει στον χρήστη την δυνατότητα να καθορίζει τα όρια της κατανάλωσης ανάλογα με τις επιθυμίες του.

3.2 Το περιβάλλον προγραμματισμού της πλατφόρμας Arduino

Για τον προγραμματισμό της πλατφόρμας Arduino έχει αναπτυχθεί μία εφαρμογή γραμμένη σε JAVA και διατίθεται για τα λειτουργικά Windows, Mac και Linux. Η εφαρμογή αυτή αποκαλείται Arduino IDE και για την συγγραφή προγραμμάτων στο συγκεκριμένο περιβάλλον χρησιμοποιείται η γλώσσα προγραμματισμού Wiring - Processing (C, C++). Τα προγράμματα που αναπτύσσει ο χρήστης ονομάζονται σκίτσα ή sketches και αποθηκεύονται ως αρχεία με επέκταση .ino. Για την μεταγλώττιση των προγραμμάτων από C, C++ στις κατάλληλες AVR εντολές της γλώσσας μηχανής το IDE χρησιμοποιεί τα εργαλεία GNU toolchain και AVR Libc. Επίσης με το εργαλείο avrdude πραγματοποιείται η φόρτωση εκτελέσιμου κώδικα στην μνήμη Flash του μικροελεγκτή που διαθέτει η πλατφόρμα Arduino. Το περιβάλλον αυτό διαθέτει αρκετά έτοιμα παραδείγματα και βιβλιοθήκες τα οποία είναι χρήσιμα για χρήστες οι οποίοι δεν είναι εξοικειωμένοι με την ανάπτυξη εφαρμογών μικροελεγκτή.

Το περιβάλλον ανάπτυξης προγραμμάτων για το Arduino που παρουσιάζεται στο σχήμα 3.1, διαθέτει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή του κώδικα, μια περιοχή μηνυμάτων, μια γραμμή εργαλείων με κουμπιά συντομεύσεων καθώς και ένα μενού επιλογών. Κατά την σύνταξη των προγραμμάτων το περιβάλλον προγραμματισμού παρέχει στον χρήστη δυνατότητες αντιγραφής/επικόλλησης και αναζήτησης/αντικατάστασης κειμένου. Στην περιοχή μηνυμάτων εμφανίζονται μηνύματα που σχετίζονται με τυχόν σφάλματα που προέκυψαν κατά τον συντακτικό έλεγχο ή την μεταγλώττιση του προγράμματος, μηνύματα επιβεβαίωσης μετά από την αποθήκευση ή φόρτωση του προγράμματος, καθώς και πληροφορίες που σχετίζονται με τον μικροελεγκτή όπως το ποσοστό μνήμης που καταλαμβάνει ο κώδικας και τα διαθέσιμα δεδομένα. Στην κάτω δεξιά γωνία του περιβάλλοντος ανάπτυξης εμφανίζεται το όνομα της πλατφόρμας που είναι συνδεδεμένη στον υπολογιστή (πχ. Arduino Uno) και η σειριακή θύρα που χρησιμοποιείται (πχ. COM3). Το μενού που υπάρχει στη γραμμή εργαλείων του Arduino IDE, επιτρέπει την δημιουργία νέων sketches, την αποθήκευση και το άνοιγμα υφιστάμενων sketches. Επίσης παρέχονται επιλογές όπως ο έλεγχος και η φόρτωση των sketches στο Arduino. Τέλος μέσω της επιλογής Serial monitor ο χρήστης έχει την δυνατότητα να ενεργοποιήσει το παράθυρο του Serial monitor για την επιβεβαίωση της λειτουργίας της εφαρμογής του.



Σχήμα 3.1- Το περιβάλλον Arduino IDE

3.2.1 Οι συντομεύσεις και τα μενού του Arduino IDE

Το περιβάλλον Arduino IDE διαθέτει μια σειρά από εικονίδια συντομεύσεων στην γραμμή εργαλείων τα οποία επιτρέπουν μια σειρά από ενέργειες όπως ο έλεγχος του προγράμματος, η φόρτωση του εκτελέσιμου κώδικα, η δημιουργία ενός νέου sketch κλπ. Τα εικονίδια συντομεύσεων και η περιγραφή της λειτουργίας τους παρουσιάζονται στον πίνακα 3.1.

Εικονίδιο	Όνομα	Εξήγηση της λειτουργίας
	Επαλήθευση (Verify)	Ελέγχει το πρόγραμμα του χρήστη για τυχόν συντακτικά λάθη (Compile).
	Φόρτωση (Upload)	Μεταγλωττίζει το πρόγραμμά και φορτώνει τον παραγόμενο εκτελέσιμο κώδικα στην αναπτυξιακή πλατφόρμα Arduino.
	Νέο (New)	Δημιουργεί ένα νέο Sketch.
	Άνοιγμα (Open)	Παρουσιάζει ένα μενού με όλα τα σκίτσα στο προεπιλεγμένο φάκελο αποθήκευσης (Sketchbook) ώστε ο χρήστης να επιλέξει το επιθυμητό sketch
	Αποθήκευση (Save)	Αποθηκεύει το τρέχον sketch.
	Σειριακή οθόνη (Serial Monitor)	Ενεργοποιεί το Serial Monitor

Πινάκας 3.1: Οι λειτουργίες της γραμμής συντομεύσεων του Arduino IDE.

Ο χρήστης έχει την δυνατότητα να επιλέξει κάποια από τις διαθέσιμες πρόσθετες επιλογές μέσω των πέντε υπο-μενού: Αρχείο (File), Επεξεργασία (Edit), Sketch, Εργαλεία (Tools), Βοήθεια (Help). Τα μενού αυτά έχουν ευαισθησία πλαισίου, δηλαδή μόνο τα στοιχεία που σχετίζονται με τις εργασίες που πραγματοποιούνται είναι ενεργοποιημένα. Το

καθένα από τα παραπάνω μενού έχει συγκεκριμένες επιλογές και λειτουργίες από τις οποίες οι πιο σημαντικές αναλύονται στην συνέχεια.

- Επεξεργασία(Edit) →**
- Copy for Forum: Αντιγράφει τον κώδικα από το σκίτσο στο πρόχειρο σε κατάλληλη μορφή προς δημοσίευση στο forum του Arduino.
 - Copy as HTML: Αντιγράφει τον κώδικα από το σκίτσο στο πρόχειρο σε μορφή HTML που είναι κατάλληλη για την ενσωμάτωση σε ιστοσελίδες.
 - Επιπλέον διαθέτει και πρόσθετες επιλογές όπως αντιγραφή, αποκοπή, αναζήτηση ,επιλογή όλων, αναίρεση κλπ.
- Σκίτσο (Sketch) →**
- Compile – Verify: Ελέγχει το σκίτσο για τυχόν σφάλματα. Επίσης υπολογίζει το χώρο μνήμης που δεσμεύει το σκίτσο και το εμφανίζει στο παράθυρο μηνυμάτων.
 - Upload: Μεταγλωττίζει το σκίτσο και φορτώνει το μεταγλωττισμένο κώδικα στην επιλεγμένη πλατφόρμα μέσω της σειριακής θύρας.
 - Show Sketch Folder: Ανοίγει τον τρέχοντα φάκελο που είναι αποθηκευμένο το σκίτσο.
 - Add File: Προσθέτει ένα νέο αρχείο στο τρέχον σκίτσο το οποίο εμφανίζεται σε μια νέα καρτέλα.
 - Include Library: Συμπεριλαμβάνει μία βιβλιοθήκη στο σκίτσο χρησιμοποιώντας την δήλωση "#include"
- Εργαλεία (Tools) →**
- Auto Format: Μορφοποιεί αυτόματα το κώδικα. Για παράδειγμα αυτόματα στοιχίζει τον κώδικα που βρίσκεται μέσα στα άγκιστρα που δηλώνουν την αρχή και το τέλος του κώδικα.
 - Archive Sketch: Αρχιεθετεί ένα αντίγραφο του τρέχοντος σκίτσου σε μορφή «.zip». Το αρχείο αυτό αποθηκεύεται στον ίδιο φάκελο με το σκίτσο.
 - Board: Επιλογή της αναπτυξιακής πλατφόρμα που θα χρησιμοποιηθεί.
 - Serial Port: Επιλογή της σειριακής θύρας μέσω της οποίας θα γίνει η σύνδεση με την διαθέσιμη αναπτυξιακή πλατφόρμα.
 - Programmer: Επιλογή του Hardware Programmer που θα χρησιμοποιηθεί για την φόρτωση του κώδικα. Συνήθως χρησιμοποιείται ο onboard programmer της επιλεγμένης αναπτυξιακής πλατφόρμας. Η επιλογή του κατάλληλου programmer είναι απαραίτητη όταν πρόκειται να φορτωθεί ο bootloader σε έναν νέο μικροελεγκτή.
 - Burn Bootloader: Η επιλογή αυτή επιτρέπει την εγγραφή ενός bootloader στο μικροελεγκτή. Αυτό δεν είναι απαραίτητο, αλλά είναι χρήσιμο στη περίπτωση που έχουμε αγοράσει ένα νέο μικροελεγκτή AT mega, ο οποίος συνήθως δεν έχει προ-εγκατεστημένο τον συγκεκριμένο bootloader.
- Help (βοήθεια) →**
- Στο παράθυρο τις Βοήθειας παρέχεται Βοήθεια για το συγκεκριμένο περιβάλλον προγραμματισμού.

3.2.2 Το Sketchbook

Το περιβάλλον Arduino IDE χρησιμοποιεί την έννοια του τετραδίου ή Sketchbook, δηλαδή έναν προεπιλεγμένο χώρο για την αποθήκευση των προγραμμάτων. Τα sketches που βρίσκονται στο φάκελο Sketchbook ανοίγουν από το μενού Αρχείο → Σχέδια (File → Sketchbook) ή από το εικονίδιο Άνοιγμα (Open) της γραμμής εργαλείων. Την πρώτη φορά που εκτελείται το Arduino IDE, αυτόματα δημιουργεί έναν κατάλογο sketchbook.

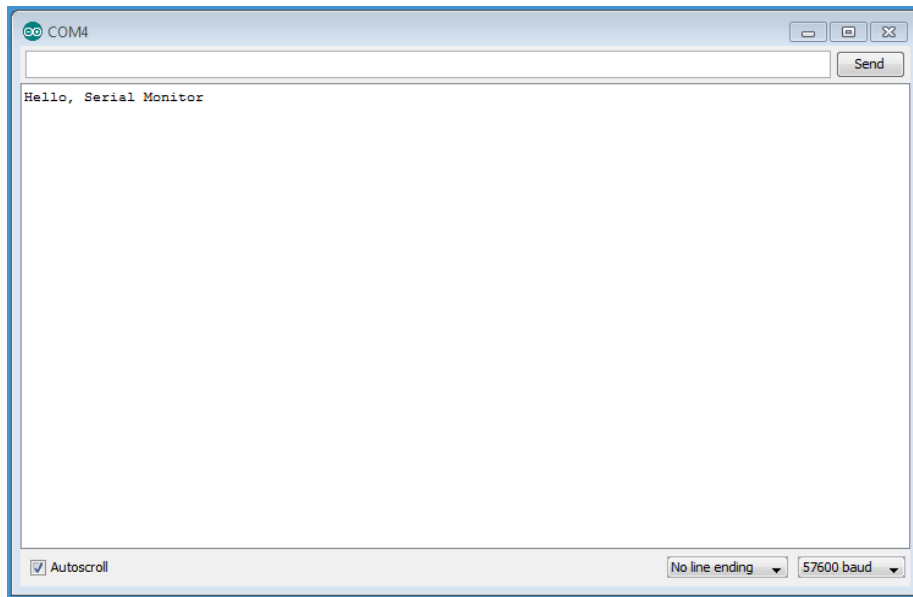
Πριν την φόρτωση του sketch, θα πρέπει να γίνει επιλογή συγκεκριμένων στοιχείων από τα μενού Εργαλεία → Πλακέτα (Tools → Board) και Εργαλεία → Σειριακή Θύρα (Tools → SerialPort). Όταν γίνει η σωστή επιλογή της σειριακής θύρας και της σωστής πλατφόρμας, ο χρήστης έχει την δυνατότητα φόρτωσης του προγράμματος στην πλατφόρμα επιλέγοντας την επιλογή upload. Η τρέχουσα πλατφόρμα Arduino θα επαναφερθεί αυτόματα (Reset) και θα αρχίσει η μεταφόρτωση (Upload). Στις περισσότερες πλατφόρμες, θα παρατηρήσουμε τα LED RX και TX να αναβοσβήνουν καθώς φορτώνεται το sketch. Το περιβάλλον Arduino IDE θα εμφανίσει ένα μήνυμα όταν η αποστολή έχει ολοκληρωθεί ή θα εμφανίσει σφάλμα εάν αυτή αποτύχει.

3.2.3 Οι βιβλιοθήκες

Οι βιβλιοθήκες παρέχουν μια σειρά από έτοιμες συναρτήσεις που αναλαμβάνουν συγκεκριμένες λειτουργίες και μπορούν να χρησιμοποιηθούν από τον χρήστη κατά την ανάπτυξη της εφαρμογής του. Οι λειτουργίες που καλύπτουν οι συναρτήσεις των βιβλιοθηκών καλύπτουν ένα μεγάλο φάσμα εφαρμογών όπως η διασύνδεση με συγκεκριμένο υλικό, η διαχείριση πολλαπλών δεδομένων, η επεξεργασία ήχου κ.α. Οι περισσότερες από τις διαθέσιμες βιβλιοθήκες δεν είναι προ-εγκατεστημένες στο Arduino IDE. Για τον λόγο αυτό ο χρήστης έχει την δυνατότητα να κατεβάσει τις βιβλιοθήκες που τον ενδιαφέρουν από την ιστοσελίδα <https://www.arduino.cc/en/Reference/Libraries>. Για να γίνει χρήση μιας βιβλιοθήκης σε ένα Sketch, πρέπει να εισαχθεί μέσω του μενού Σχέδιο → Εισαγωγή Βιβλιοθήκης (Sketch → Import Library). Η παραπάνω ενέργεια θα εισάγει μια ή περισσότερες δηλώσεις «#include» στην αρχή του κώδικα και έτσι οι συναρτήσεις της συγκεκριμένης βιβλιοθήκης θα μεταγλωττιστούν μαζί με το πρόγραμμά του χρήστη. Επειδή οι βιβλιοθήκες φορτώνονται και αυτές μαζί με το sketch στη πλατφόρμα, αυξάνουν τη ποσότητα του χώρου που καταλαμβάνει το πρόγραμμα στη μνήμη του Arduino. Για τον λόγο αυτό θα πρέπει να συμπεριλαμβάνονται σε ένα πρόγραμμα μόνο οι βιβλιοθήκες που είναι απαραίτητες.

3.2.4 Serial Monitor

Το Arduino IDE διαθέτει την σειριακή οθόνη (Serial Monitor) που παρουσιάζεται στο σχήμα 3.2, η οποία αναλαμβάνει την απεικόνιση των δεδομένων που αποστέλλονται από το Arduino μέσω της σειριακής θύρας. Για να γίνει αποστολή δεδομένων στη πλατφόρμα Arduino μέσω του Serial monitor θα πρέπει ο χρήστης να εισάγει το επιθυμητό μήνυμα στο αντίστοιχο πεδίο και να επιλέξει το κουμπί Αποστολή (Send) ή πατήσει το πλήκτρο Enter. Επίσης ο ρυθμός μεταφοράς δεδομένων (baud rate) που έχει επιλεγεί στο Serial monitor και αυτός που έχει δηλωθεί στον κώδικά θα πρέπει να είναι ίδιοι. Σημειώνεται ότι σε λειτουργικά συστήματα Mac ή Linux, η αναπτυξιακή πλατφόρμα Arduino κάνει επανεκκίνηση, όταν συνδέεται με το Serial monitor.



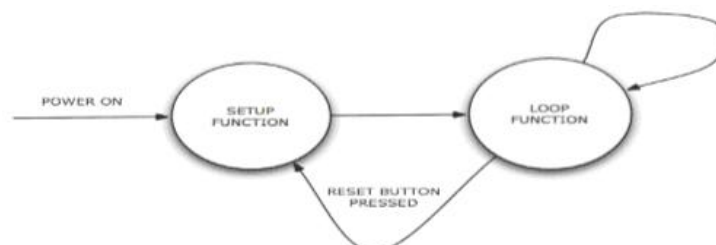
Σχήμα 3.2: Serial Monitor

3.2.5 Η δομή του κώδικα

Η δομή του προγράμματος το οποίο αναπτύσσεται στο περιβάλλον προγραμματισμού του Arduino παρουσιάζεται στο πρόγραμμα 3.1. Όπως παρατηρείται το πρόγραμμα αποτελείται από τρία βασικά μέρη, το πρώτο μέρος όπου γίνεται η δήλωση των γενικών μεταβλητών (Global variables) και των συμπεριλαμβανόμενων βιβλιοθηκών. Το δεύτερο μέρος όπου γίνονται οι απαραίτητες αρχικοποιήσεις (συνάρτηση setup), και το τρίτο μέρος όπου γράφεται το κυρίως πρόγραμμα (συνάρτηση loop).

```
//δήλωση παγκοσμίων μεταβλητών
Void setup()
{
//αρχικοποιήσεις
}
Void loop()
{
//Κώδικας
}
```

Πρόγραμμα 3.1 – Δομή του προγράμματος.



Σχήμα 3.3: Διάγραμμα ροής του προγράμματος.

Όλα τα προγράμματα που γράφονται πρέπει να είναι της παραπάνω μορφής και πρέπει οπωσδήποτε να περιλαμβάνει και τα τρία μέρη ακόμα και αν σε κάποιο από αυτά δεν είναι γραμμένο κάποια δήλωση ,αρχικοποίηση ή εντολή.

Για να γίνει κατανοητό κάποιο μέρος του προγράμματος ή κάποιας εντολής μέσα στο πρόγραμμα υπάρχει η δυνατότητα να γραφτούν σχόλια ή κομμάτια κειμένου που επεξηγούν την λειτουργία του προγράμματος. Το τμήμα του κώδικα που βρίσκεται μεταξύ των συμβόλων `/*` και `*/` ή το κείμενο που βρίσκονται μετά από το σύμβολο `/**` (πρόγραμμα 3.2) αποτελούν σχόλια του κώδικα και δεν λαμβάνονται υπόψη κατά την μεταγλώττιση και την φόρτωση του εκτελέσιμου κώδικα στον μικροελεγκτή.

```
/*  
Analog input, analog output, serial output  
Reads an analog input pin, maps the result to a range from 0 to 255  
and uses the result to set the pulsewidth modulation (PWM) of an output pin.  
Also prints the results to the serial monitor.  
This example code is in the public domain.  
*/  
  
pinMode (ledPin,OUTPUT); // set the ledPin as output.
```

Πρόγραμμα 3.2: Σχόλια

Γενικές και τοπικές μεταβλητές

Το πρώτο από τα τρία μέρη του κώδικα είναι το κομμάτι όπου γίνεται η δήλωση των γενικών μεταβλητών και των συμπεριλαμβανομένων βιβλιοθηκών. Οι γενικές μεταβλητές χρησιμοποιούνται για την αποθήκευση δεδομένων και είναι διαθέσιμες σε όλο το πρόγραμμα. Δηλαδή μπορούν να αξιοποιηθούν σε οποιοδήποτε συνάρτηση που περιλαμβάνεται στο πρόγραμμα σε αντίθεση με τις τοπικές μεταβλητές οι οποίες ορίζονται μέσα σε κάποια συνάρτηση με αποτέλεσμα η χρήση τους να περιορίζεται μέσα στην συνάρτηση αυτή. Οι τύποι δεδομένων που μπορούν να αποθηκευτούν σε μια μεταβλητή είναι ακέραιοι αριθμοί με ή χωρίς πρόσημο, αριθμοί κινητής υποδιαστολής, σειρές αλφαριθμητικών χαρακτήρων κλπ.

Συναρτήσεις

Η δήλωση Void πριν από το όνομα της συνάρτησης (πχ setup) δηλώνει ότι αυτή η συνάρτηση δεν επιστρέφει κάποιο αποτέλεσμα. Τα τυχόν ορίσματα της συνάρτησης αναγράφονται μέσα στις παρενθέσεις που ακολουθούν το όνομα της συνάρτησης. Στην περίπτωση της συνάρτησης setup δεν υπάρχουν ορίσματα και για τον λόγο αυτό μετά το όνομα της συνάρτησης ακολουθούν μόνο οι παρενθέσεις. Ο κώδικας της συνάρτησης βρίσκεται μέσα σε αγκύλες και συχνά καλείται ως σώμα της συνάρτησης. Υπάρχουν συναρτήσεις οι οποίες είναι έτοιμες και παρέχονται από την ομάδα σχεδιασμού του Arduino αλλά υπάρχει και η δυνατότητα ο χρήστης δημιουργήσει τις δικές του συναρτήσεις ανάλογα με τις ανάγκες του. Για την σωστή λειτουργία του προγράμματος που αναπτύσσεται με το Arduino IDE πρέπει να συμπεριλαμβάνονται οπωσδήποτε οι δυο βασικές συναρτήσεις, η συνάρτηση setup() και η συνάρτηση loop().

Συνάρτηση setup()

Η συνάρτηση setup() εκτελείται στην αρχή του προγράμματος και μόνο την πρώτη φορά. Χρησιμοποιείται για τις αρχικοποιήσεις ιδιοτήτων, βιβλιοθηκών και διαφόρων μεταβλητών. Οι μεταβλητές που δηλώνονται στην συνάρτηση setup() δεν είναι προσβάσιμες από άλλες συναρτήσεις, συμπεριλαμβανομένης και της loop(). Στην συνάρτηση setup()

πραγματοποιείται επίσης η αρχικοποίηση των συσκευών που είναι συνδεδεμένες με το Arduino όπως η οθόνη, το Ethernet Shield κλπ.

Συνάρτηση loop()

Η συνάρτηση loop() εκτελείται αμέσως μετά από την συνάρτηση setup(). Στην συνάρτηση loop() γράφεται το κύριο σώμα του προγράμματος. Όπως παρατηρείτε και από το όνομα της συνάρτησης ο κώδικας που περιλαμβάνεται σε αυτή την συνάρτηση επαναλαμβάνεται συνεχώς. Επομένως ο μικροελεγκτής εκτελεί την συνάρτηση loop μέχρι να πατηθεί το κουμπί RESET ή να διακοπεί τάση τροφοδοσίας του μικροελεγκτή.

3.2.6 Παράδειγμα ενός ολοκληρωμένου προγράμματος

Παρακάτω ακολουθεί ένα παράδειγμα(πρόγραμμα 3.3) που περιλαμβάνει όλες τις παραπάνω βασικές έννοιες. Στο sketch αυτό υλοποιείται μια εφαρμογή που ανάβει και σβήνει ένα led κάθε ένα δευτερόλεπτο. Τα αντικείμενα που είναι απαραίτητα για την υλοποίηση του παραδείγματος είναι μια πλακέτα Arduino ένα LED και μια αντίσταση όπως παρουσιάζεται στο σχήμα 3.4.



Σχήμα 3.4: Ένα Arduino Uno συνδεδεμένο με ένα led.

```
/*  
Example:  
Turns on an LED on for one second, then off for one second, repeatedly.  
*/  
void setup()// the setup function runs once when you press reset or power the board  
{  
  pinMode(13, OUTPUT); // initialize digital pin 13 as an output  
}  

```

Πρόγραμμα 3.3: παράδειγμα blink

Εδώ παρατηρείται ότι έχει ορισθεί ο ακροδέκτης 13 για να χρησιμοποιηθεί ως έξοδος μέσω της συνάρτησης `pinMode(13, OUTPUT)`. Αυτό συμβαίνει μέσα στην συνάρτηση `setup()` η οποία όπως αναφέρθηκε εκτελείται μόνο μια φορά όταν ξεκινάει το Arduino. Παρακάτω ορίζεται η συνάρτηση `loop()` στην οποία παρατηρείται ανάθεση στον ακροδέκτη 13 της τάση των 5V με την συνάρτηση `digitalWrite(13, HIGH)`. Με την χρήση της συνάρτησης `delay(1000)` διατηρείται η τάση στον ακροδέκτη 13 για ένα δευτερόλεπτο. Με την συνάρτηση `digitalWrite(13, LOW)` αφαιρείται από τον ακροδέκτη η τάση 5V για επίσης ένα δευτερόλεπτο. Η συνάρτηση `loop()` εκτελείται διαρκώς για όση ώρα είναι σε λειτουργία η πλακέτα του Arduino.

3.3 Εισαγωγή στο πρόγραμμα LabVIEW.

Το LabVIEW, είναι ένα γραφικό περιβάλλον προγραμματισμού που αναπτύχθηκε από την National Instruments. Η γλώσσα προγραμματισμού που χρησιμοποιεί, ονομάζεται G (Graphical Language) και είναι ένα είδος γλώσσας VPL (Visual Programming Language), η οποία δίνει την δυνατότητα στον χρήστη να προγραμματίζει, όχι με βάση την γραφή εντολών αλλά με τον χειρισμό γραφικών εικονιδίων.

Τα προγράμματα που δημιουργούνται στο LabVIEW, ονομάζονται VIs (Virtual Instruments). Κάθε VI έχει δύο μέρη: το block diagram και το front panel. Το front panel είναι το περιβάλλον το οποίο χειρίζεται ο χρήστης, χρησιμοποιώντας ελεγκτές (controls) και ενδείκτες (indicators). Μέσω των controls ο χρήστης έχει την δυνατότητα να μεταβάλλει τις παραμέτρους του προγράμματος ενώ μέσω των indicators να παρατηρεί τα αποτελέσματα του προγράμματος. Το block diagram, είναι το περιβάλλον στο οποίο συντάσσεται ο γραφικός κώδικας. Όλα τα controls και τα indicators παρουσιάζονται στο block diagram ως τερματικά εικονίδια (terminals). Το περιβάλλον του block diagram περιέχει τις απαιτούμενες δομές και συναρτήσεις που απαιτούνται για την δημιουργία του κώδικα μιας εφαρμογής. Μέσω του κώδικα επιτυγχάνεται η επεξεργασία των αρχικών δεδομένων, που περιέχουν τα controls του front panel, ενώ στην συνέχεια παρουσιάζονται τα τελικά δεδομένα σε κάποια indicators.

Επίσης, ο γραφικός κώδικας του LabVIEW, διευκολύνει την κατανόηση και την ανάπτυξη ενός προγράμματος. Η μεγάλη ποικιλία των έτοιμων παραδειγμάτων που παρέχει το LabVIEW συμβάλει σημαντικά στην εξοικείωση ενός νέου χρήστη με την ανάπτυξη διαφόρων εφαρμογών. Επιπλέον, είναι δυνατή η δημιουργία VIs, τα οποία θα έχουν την δυνατότητα να επικοινωνούν μεταξύ τους τόσο σε τοπικό επίπεδο όσο και απομακρυσμένα μέσω διαδικτύου.

Στις παραγράφους που ακολουθούν θα γίνει μια σύντομη παρουσίαση στο περιβάλλον του LabVIEW 2011. Επίσης μέσω παραδειγμάτων θα παρουσιαστεί ο τρόπος με τον οποίο μπορούν να δημιουργηθούν απλά VIs.

3.3.1 Το περιβάλλον του LabVIEW 2011.

Η εκκίνηση του LabVIEW πραγματοποιείται είτε επιλέγοντας την συντόμευση του προγράμματος που βρίσκεται στην επιφάνεια εργασίας, είτε επιλέγοντας στην Έναρξη → Όλα τα προγράμματα → National Instruments → LabVIEW 2011. Όταν ολοκληρωθεί η εκκίνηση του LabVIEW θα εμφανιστεί το αρχικό παράθυρο της εφαρμογής που παρουσιάζεται στο σχήμα 3.5.

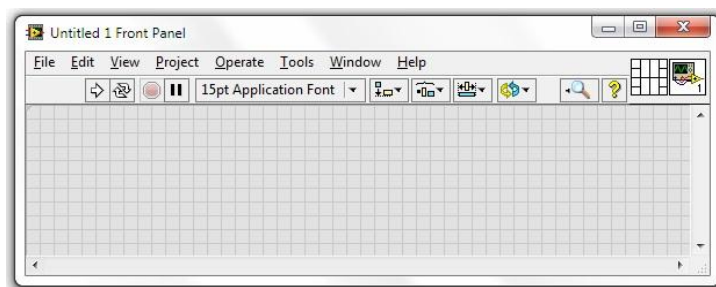


Σχήμα 3.5: Το παράθυρο Getting Started του LabVIEW 2011.

Σε αυτό το παράθυρο εμφανίζονται οι αρχικές λειτουργίες του LabVIEW. Στο δεξιό μέρος του παραθύρου, υπάρχουν οι κατηγορίες: α) Latest from ni.com όπου δίνεται η δυνατότητα στον χρήστη να μάθει μέσω της ni.com, νέες πληροφορίες που αφορούν το πρόγραμμα και γενικά για την National Instruments, β) online support, δηλαδή προσφέρεται η είσοδος σε κάποια forum ή κάποια ιστοσελίδα υποστήριξης και γ) help, όπου ο χρήστης μπορεί να βρει κάποια παραδείγματα, να κατεβάσει (download) κάποιους drivers που ίσως να χρειάζεται καθώς και να εγκαταστήσει ορισμένα Add-ons.

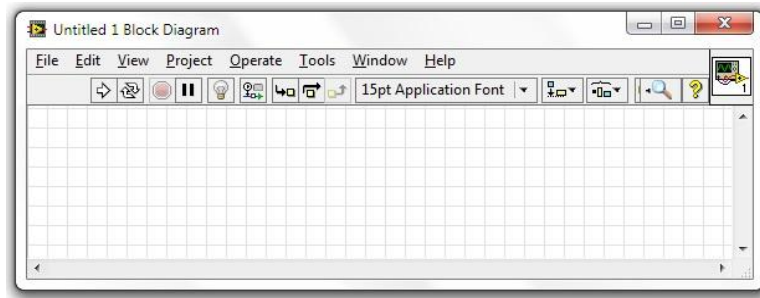
Στο αριστερό μέρος του παραθύρου, υπάρχουν οι κατηγορίες: α) New, από όπου μπορεί να γίνει η δημιουργία ενός νέου VI ή project και β) Open, στο οποίο επισημαίνονται τα VIs που είχαν ενεργοποιηθεί πρόσφατα ή να ανοίξει ένα υπάρχον VI.

Η έναρξη ενός κενού VI πραγματοποιείται: πατώντας από την κατηγορία new → blank VI ή μέσω file → new VI. Με την έναρξη ενός καινούριου VI, εμφανίζονται δύο παράθυρα εκ των οποίων, το πρώτο παράθυρο που εμφανίζεται είναι το front panel (σχήμα 3.6).



Σχήμα 3.6: Front panel ενός καινούριου VI.

Το δεύτερο παράθυρο που εμφανίζεται, είναι το block diagram (σχήμα 3.7).



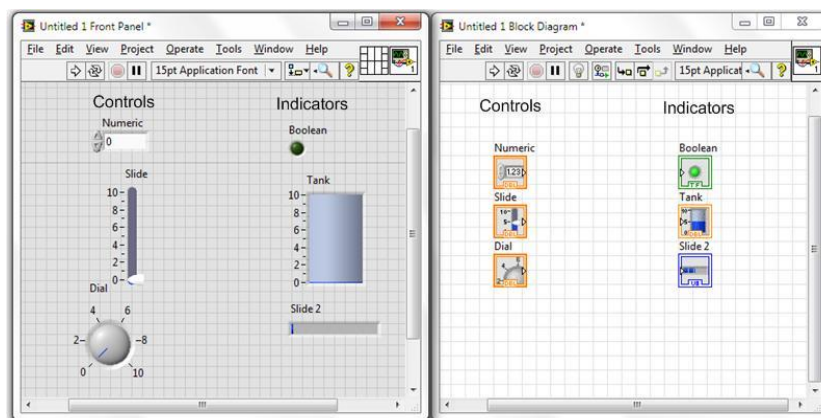
Σχήμα 3.7: Block diagram ενός καινούριου VI

Λειτουργία του Front panel.

Στο παράθυρο του Front Panel, δημιουργείται το front panel ενός VI το οποίο είναι το περιβάλλον αλληλεπίδρασης (interface) του VI με τον χρήστη. Το Front Panel παρέχει στον χρήστη, για την διαμόρφωση του interface, μία παλέτα εργαλείων και αντικειμένων που ονομάζεται Controls Palette. Επίσης το παράθυρο του Front Panel διαθέτει μια μπάρα εργαλείων μέσω της οποίας δίνεται η δυνατότητα, να ξεκινά ή να σταματά η εκτέλεση του VI, να καθορίζεται το φόντο και η γραμματοσειρά των κειμένων που θα περιέχει το front panel, καθώς και η στοίχιση των αντικειμένων του.

Λειτουργία του Block Diagram.

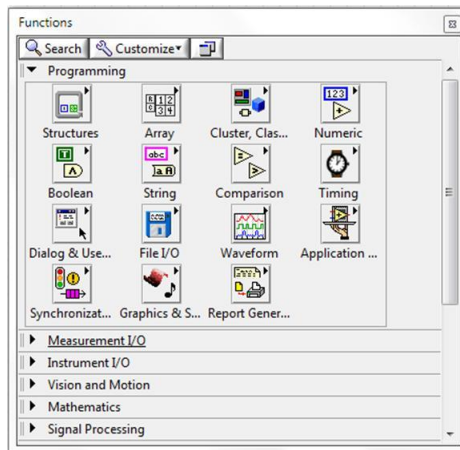
Το Block Diagram, είναι το παράθυρο στο οποίο ο χρήστης δημιουργεί τον κώδικα ενός VI. Τα τερματικά εικονίδια (terminals) των controls και των indicators έχουν στόχο να μεταφέρουν τα δεδομένα από το Block Diagram στο Front Panel και αντίστροφα. Στην περίπτωση που ανοιχθεί ένα VI και δεν εμφανίζεται το Block Diagram ο χρήστης θα πρέπει να επιλέξει μέσω του Front Panel, Windows → Show Block Diagram. Στο σχήμα 3.8 απεικονίζεται το Front Panel και το Block Diagram ενός VI με τυχαία controls και τα indicators.



Σχήμα 3.8: Παρουσίαση controls-indicators στο Front Panel και στο Block Diagram.

Στο Block Diagram εμφανίζεται η Functions Palette (σχήμα 3.9), η οποία διαθέτει όλα τα απαραίτητα αντικείμενα και συναρτήσεις για την ανάπτυξη του κώδικα. Επίσης το Block Diagram περιέχει μια μπάρα λειτουργιών αντίστοιχη με του Front Panel, με την οποία μπορεί να γίνει ο χειρισμός του VI.

Η Functions Palette του Block Diagram.

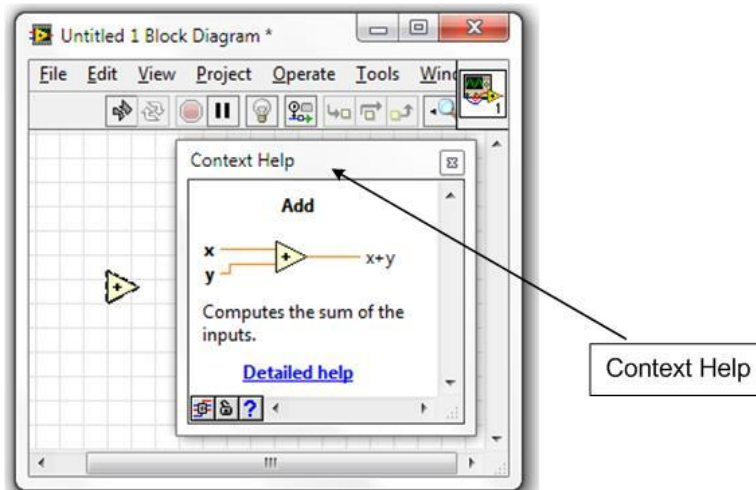


Σχήμα 3.9: Functions Palette του Block Diagram.

Κάθε συνάρτηση εισάγεται στο Block Diagram με την διαδικασία drag and drop, κάνοντας έτσι εύκολη την χρήση της. Για την ευκολότερη εύρεση των αντικειμένων, η Functions Palette παρέχει την δυνατότητα search, με την οποία μπορεί να γίνει η εύρεση μίας συνάρτησης μέσω της ονομασίας της. Επίσης, μέσω της επιλογής customize μπορεί να διαμορφωθεί η εμφάνιση της παλέτας. Η εμφάνιση της Functions Palette στο Block Diagram, γίνεται με δεξί-κλικ του ποντικιού ή επιλέγοντας View → Functions Palette.

Help – Context Help.

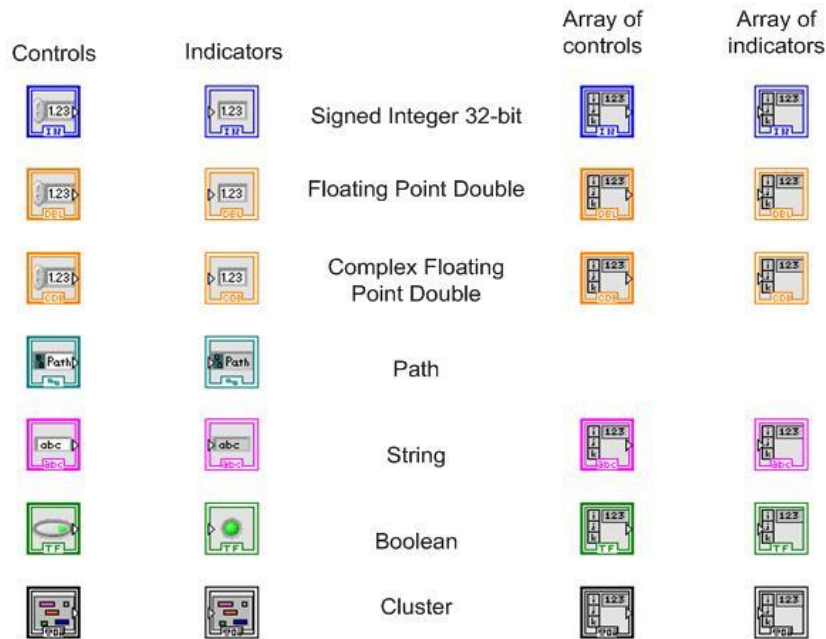
Τέλος, το LabVIEW παρέχει βιβλιοθήκες οι οποίες προσφέρουν λεπτομερείς οδηγίες για κάθε μία λειτουργία. Για την εύρεση των πληροφοριών αυτών, είτε στο Block Diagram είτε στο Front Panel, ο χρήστης μπορεί να επιλέξει Help → LabVIEW Help. Μπορεί επίσης να βρει και άλλα βοηθήματα στο Help, ένα εκ των οποίων είναι το Context Help (σχήμα 3.10) το οποίο όταν ενεργοποιείται παρουσιάζει πληροφορίες σχετικά με το αντικείμενο ή την συνάρτηση που υποδεικνύει ο κέρσορας.



Σχήμα 3.10: Το παράθυρο Context Help.

3.3.2 Οι τύποι δεδομένων.

Οι τύποι δεδομένων που υποστηρίζει το LabVIEW παρουσιάζονται στο σχήμα 3.11.



Σχήμα 3.11: Τύποι δεδομένων.

3.3.3 Οι συναρτήσεις του LabVIEW.

Οι βασικές συναρτήσεις του LabVIEW είναι ομαδοποιημένες στη Functions Palette σύμφωνα με το είδος των δεδομένων που διαχειρίζονται και σύμφωνα με την λειτουργία τους. Επομένως οι παραπάνω συναρτήσεις ομαδοποιούνται στις εξής κατηγορίες:

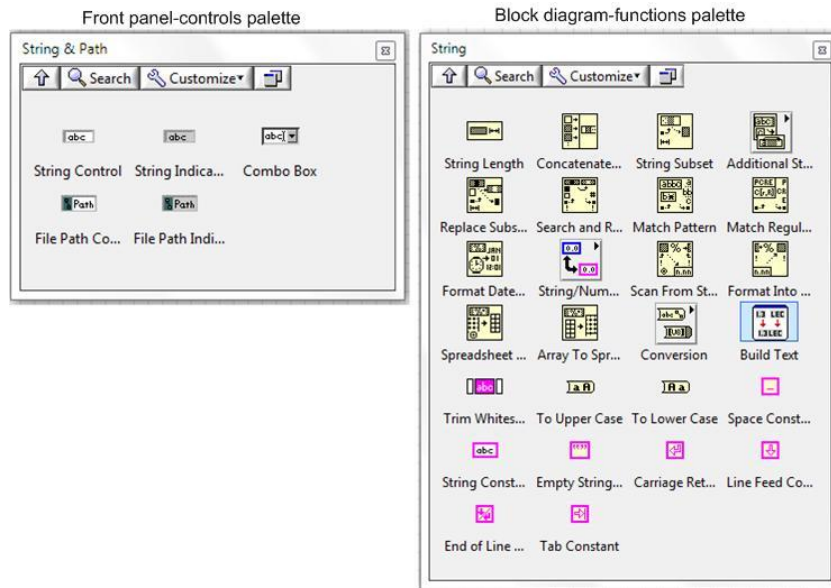
1. Strings.
2. Arrays.
3. Clusters.
4. Loops.
5. Case Structures.
6. Sequence Structure.
7. Local variables.
8. Γραφικές Παραστάσεις.
9. SubVIs.

Strings

Τα Strings είναι ακολουθίες των προβαλλόμενων ή μη-προβαλλόμενων χαρακτήρων ASCII (American Standard Code for Information Interchange). Μερικές από τις πιο κοινές εφαρμογές των Strings είναι οι ακόλουθες:

1. Δημιουργία ενός απλού γραπτού μηνύματος.
2. Έλεγχος οργάνων ή εξωτερικών διατάξεων, στέλνοντας εντολές στο όργανο ή την εξωτερική διάταξη.
3. Αποθήκευση δεδομένων στον δίσκο
4. Πληροφόρηση του χρήστη μέσω παραθύρων διαλόγου.

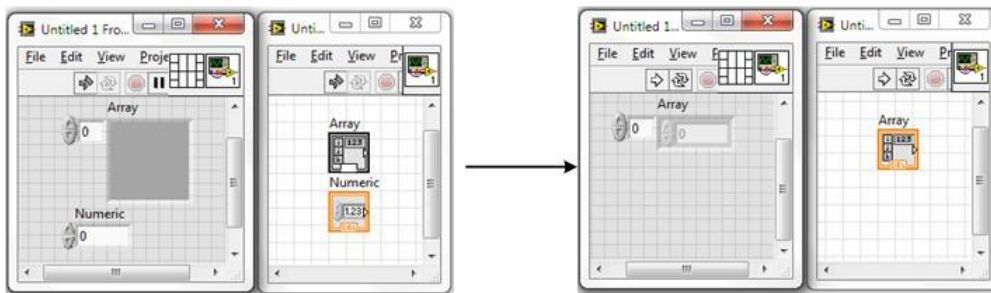
Στο σχήμα 3.12 παρουσιάζονται τα αντικείμενα τύπου String που περιέχει η Controls Palette και οι συναρτήσεις που διαθέτει η Function Palette για την διαχείριση των Strings.



Σχήμα 3.12: Συναρτήσεις Strings στην Controls & Function Palette.

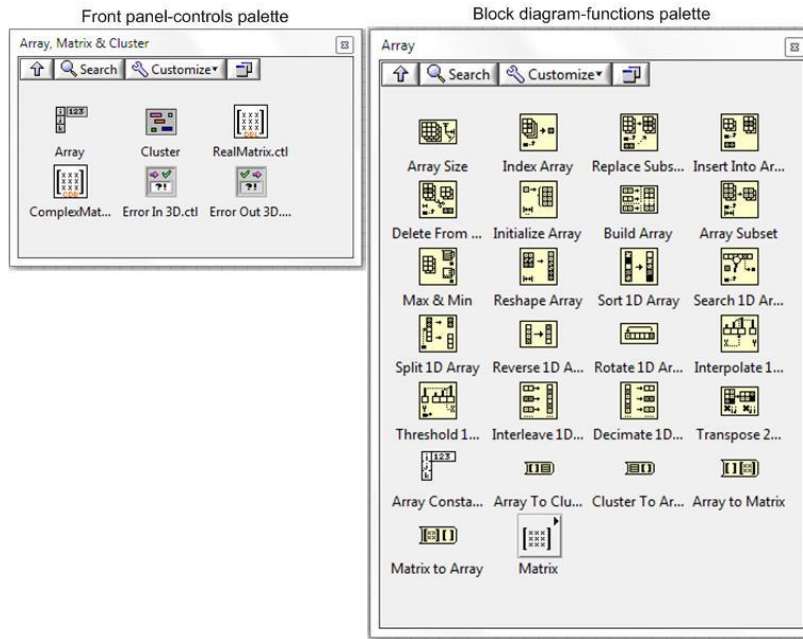
Arrays

Οι πίνακες - Arrays ομαδοποιούν στοιχεία δεδομένων του ίδιου τύπου. Ένας πίνακας μπορεί να έχει περισσότερες από μια διαστάσεις. Στο σχήμα 4.13 παρουσιάζεται η δημιουργία ενός μονοδιάστατου πίνακα που περιέχει αριθμητικά δεδομένα τύπου floating point double (δεδομένα διπλής ακρίβειας κινητής υποδιαστολής).



Σχήμα 3.13: Η δημιουργία ενός πίνακα με αριθμητικά δεδομένα.

Η δημιουργία ενός πίνακα στο Front Panel επιτυγχάνεται με την εισαγωγή του αντίστοιχου αντικειμένου (Controls Palette → Modern → Array, Matrix & Cluster). Ο πίνακας Array που σχηματίστηκε, μπορεί να αποτελείται μόνο από controls ή μόνο από indicators. Στο σχήμα 3.15 παρουσιάζονται τα αντικείμενα τύπου Array που περιέχει η Controls Palette και οι συναρτήσεις που διαθέτει η Function Palette για την διαχείριση των πινάκων.



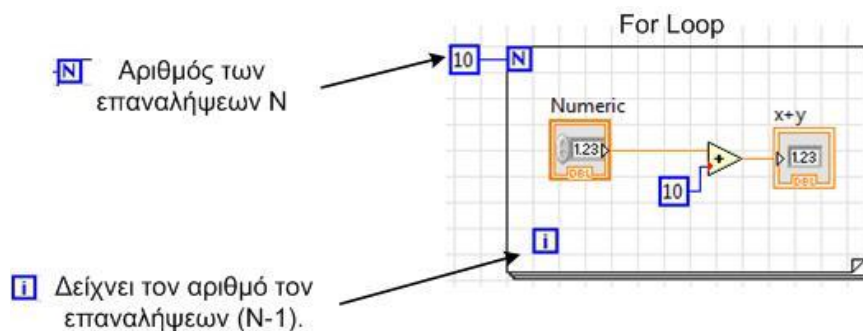
Σχήμα 3.15: Συναρτήσεις Array στην Controls & Function Palette.

Loops

Οι βρόχοι επανάληψης-loops, είναι κάποιες δομές-structures με τις οποίες επιτυγχάνεται η επανάληψη μιας συγκεκριμένης διαδικασίας σε ένα VI. Οι επαναλήψεις αυτές δεν είναι ίδιες, αλλά καθορίζονται από το είδος του βρόχου επανάληψης. Με την δημιουργία ενός βρόχου επανάληψης, μπορούν να προστεθούν περαιτέρω καινούρια στοιχεία. Τα είδη των βρόχων επανάληψης είναι οι For Loop, While Loop και ηTimed Loop.

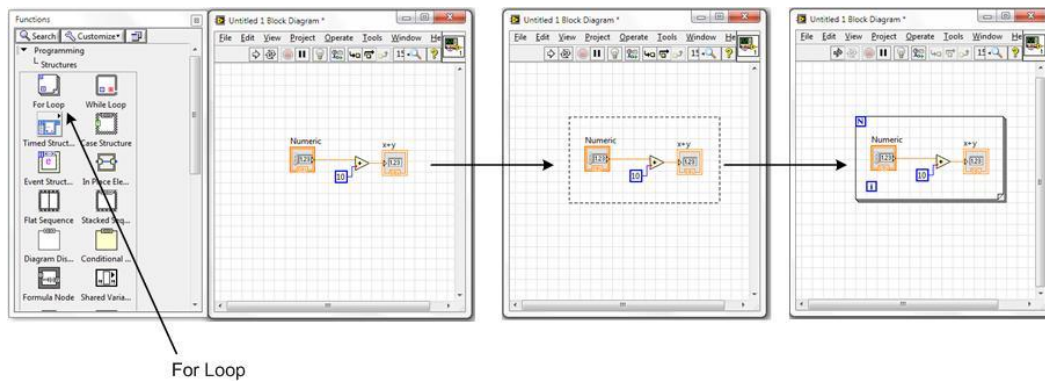
For Loop

Είναι ο βρόχος επανάληψης (σχήμα 3.15), όπου ο αριθμός των επαναλήψεων (N) καθορίζεται από τον αριθμό που θα του δώσει ο χρήστης. Όταν οι επαναλήψεις αυτές ισοδυναμούν με το όριο που έχει δοθεί, η For Loop σταματά.



Σχήμα 3.15: For Loop.

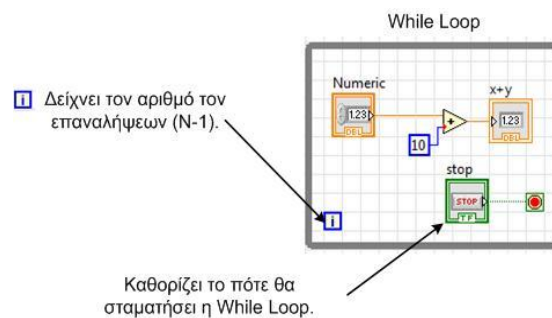
Η διαδικασία για την δημιουργία ενός For Loop παρουσιάζεται στο σχήμα 3.16 και η εισαγωγή της στο Block Diagram επιτυγχάνεται μέσω της Functions Palette → Programming → Structures → For Loop.



Σχήμα 3.16: Δημιουργία μιας For Loop.

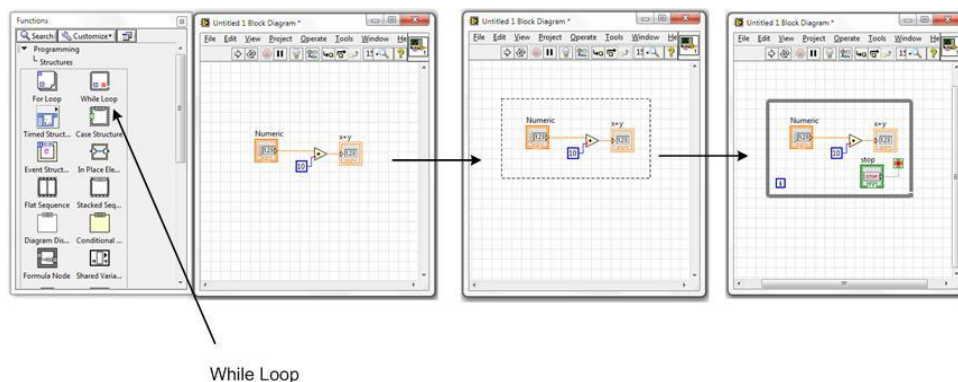
While Loop

Η While Loop (σχήμα 3.17), είναι ο βρόχος επανάληψης, όπου επαναλαμβάνει ένα sub diagram. Σε σχέση με την For Loop, στην While Loop δεν δίνεται κάποιος ορισμένος αριθμός επαναλήψεων με αποτέλεσμα να επαναλαμβάνει ένα sub diagram συνέχεια, μέχρι να ικανοποιηθεί η αντίστοιχη συνθήκη.



Σχήμα 3.17: While Loop.

Η διαδικασία για την δημιουργία ενός While Loop παρουσιάζεται στο σχήμα 3.18 και η εισαγωγή της στο Block Diagram επιτυγχάνεται μέσω της Functions Palette → Programming → Structures → While Loop.



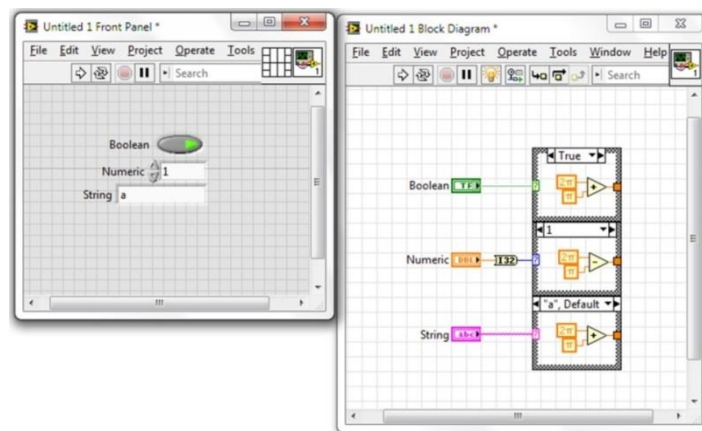
Σχήμα 3.18: Δημιουργία μιας While Loop.

Timed Loop:

Αυτή η περίπτωση του βρόχου επανάληψης, χρησιμοποιείται, όταν οι επανάληψη ενός sub diagram ή περισσότερων, πρέπει να γίνει εντός κάποιων συγκεκριμένων χρονικών ορίων. Στην Timed Loop δεν καθορίζεται ο αριθμός των επαναλήψεων, αλλά το χρονικό διάστημα που διαρκεί κάθε επανάληψη καθώς και η συχνότητα των επαναλήψεων. Επίσης θα πρέπει να προσδιοριστεί και μια σειρά άλλων παραμέτρων όπως η χρονική στιγμή που ολοκληρωθεί η Timed Loop.

Case Structures

Οι Case Structures, είναι δομές οι οποίες αποτελούνται από δύο η περισσότερες περιπτώσεις (cases). Κάθε περίπτωση περιλαμβάνει τον κώδικα που θα εκτελεστεί εφόσον επιλεγεί η συγκεκριμένη περίπτωση μέσω της εισόδου Case selector. Ο τύπος δεδομένων που μπορεί να συνδεθεί με την Case selector, μπορεί να είναι boolean, string, integer, enumerated type ή error cluster. Ανάλογα με το είδος της εισόδου, θα αλλάξει και το χρώμα της Case selector (σχήμα3.19).



Σχήμα 3.19: Οι διαφορετικές περιπτώσεις για την επιλογή της case που θα εκτελεστεί σε μια Case Structure.

Η περίπτωση Default εκτελείται όταν η τιμή του Case selector είναι ίση με την τιμή που αντιστοιχεί στην Default περίπτωση ή όταν η τιμή του Case selector δεν αντιπροσωπεύει κάποια από τις διαθέσιμες περιπτώσεις της δομής Case. Η εισαγωγή μιας νέας case γίνεται με δεξί κλικ στην Case Structure και επιλέγοντας add case after ή add case before. Η εισαγωγή μιας νέας Case Structure στο Block Diagram επιτυγχάνεται μέσω της Functions Palette → Programming → Structures → Case Structure.

Γραφικές παραστάσεις

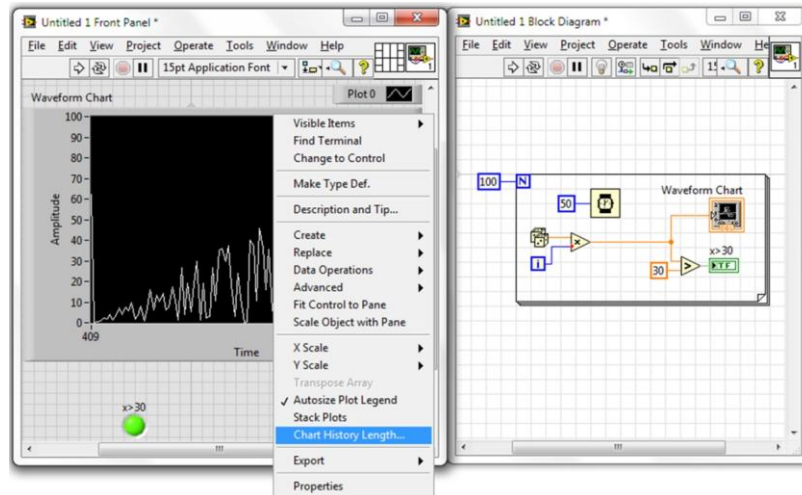
Πολλές φορές κατά την υλοποίηση μιας εφαρμογής απαιτείται η γραφική αναπαράσταση των δεδομένων της. Τα βασικά αντικείμενα που διαθέτει το LabVIEW για την γραφική αναπαράσταση των δεδομένων είναι:

1. Waveform Chart.
2. Waveform Graph.
3. XY Graph.

Waveform chart

Το Waveform chart είναι ένα είδος αριθμητικού ενδείκτη-indicator, όπου εμφανίζει ένα ή περισσότερα διαγράμματα δεδομένων. Το Waveform chart έχει την δυνατότητα να διατηρεί ένα ιστορικό δεδομένων, το οποίο προβάλλει στην οθόνη του, προσθέτοντας κάθε

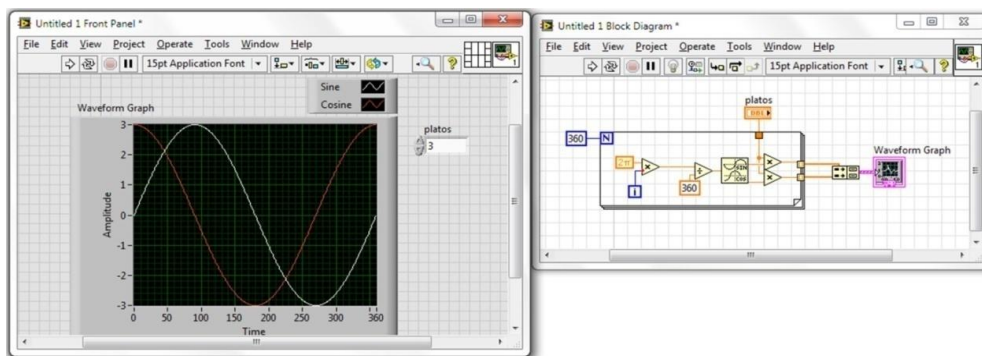
φορά και από ένα στοιχείο. Το μέγεθος του ιστορικού που θα προβάλει, ορίζεται από το Chart History Length, το οποίο διαμορφώνεται από τον χρήστη επιλέγοντας με δεξί κλικ το Waveform chart. Το Chart History Length έχει ως προεπιλογή 1024 (data points). Η συχνότητα με την οποία στέλνονται τα δεδομένα στο Waveform chart, καθορίζει την συχνότητα επανασχεδιασμού του γραφήματος. Στο σχήμα 3.20 παρουσιάζεται ένα παράδειγμα με την χρήση του Waveform chart.



Σχήμα 3.20: Waveform Chart.

Waveform Graph

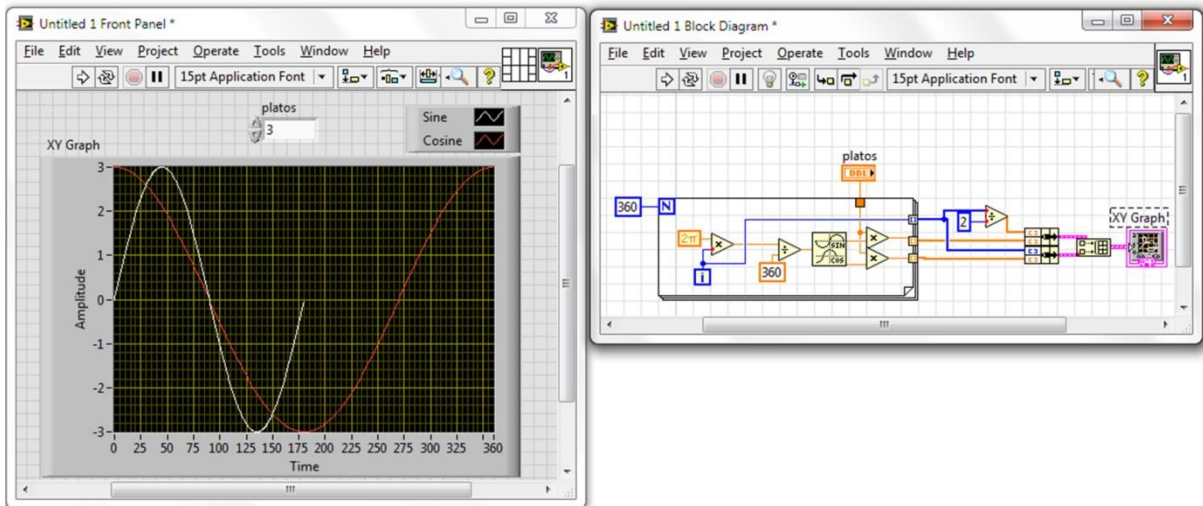
Το Waveform Graph είναι ένα είδος αριθμητικού ενδείκτη-indicator, το οποίο εμφανίζει ένα ή περισσότερα γραφήματα, τα οποία έχουν ομοιόμορφη δειγματοληψία μετρήσεων. Τα δεδομένα αυτά, το Waveform Graph τα δέχεται από έναν ολοκληρωμένο πίνακα, ενώ το προβαλλόμενο γράφημα ανανεώνεται μόνο όταν όλα τα δεδομένα είναι διαθέσιμα. Στο σχήμα 3.21 απεικονίζεται ένα παράδειγμα ενός Waveform Graph.



Σχήμα 3.21: Waveform Graph. (sine-cosine).

XY Graph

Το XY Graph είναι ένα είδος αριθμητικού ενδείκτη-indicator, το οποίο σχεδιάζει ένα ή περισσότερα γραφήματα από ένα Cluster που περιέχει δύο πίνακες με δεδομένα, εκ των οποίων ο ένας περιέχει τα στοιχεία του άξονα X και ο άλλος τα στοιχεία του άξονα Y. Μέσω του XY Graph, μπορεί να προβληθεί οποιαδήποτε γραφική παράσταση, ανεξαιρέτως εάν έχουν ομοιόμορφη δειγματοληψία ή όχι, ενώ το προβαλλόμενο γράφημα ανανεώνεται μόνο όταν όλα τα δεδομένα είναι διαθέσιμα. Στο σχήμα 3.22 απεικονίζεται ένα παράδειγμα με το XY Graph.

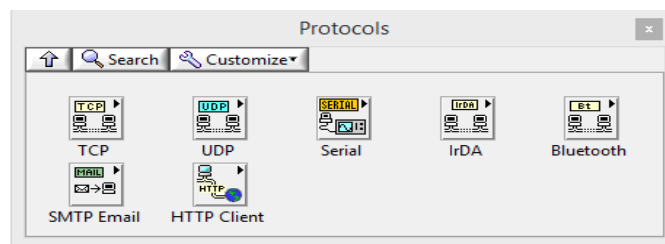


Σχήμα 3.22: XY Graph (sine-cosine).

Μέσω αυτών των Graphs, μπορεί ο χρήστης να αναπαραστήσει οποιαδήποτε γραφήματα επιθυμεί, δίνοντας του την δυνατότητα να μπορεί να τα επεξεργαστεί όπως θέλει αυτός. Η δημιουργία των Graphs αυτών, γίνεται στο Front Panel μέσω της Controls Palette → Graph

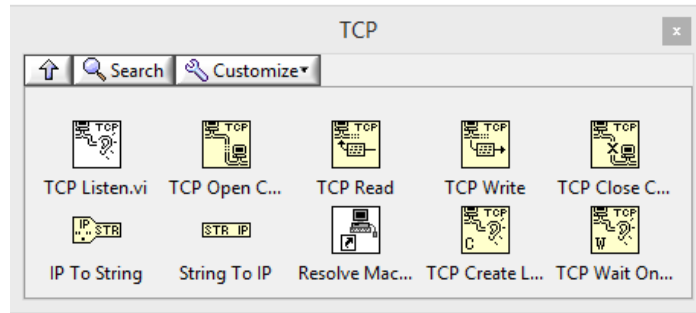
3.3.4 Πρωτόκολλα επικοινωνίας

Τα πρωτόκολλα IP (Internet Protocol), UDP (User Datagram Protocol), και TCP (Transmission Control Protocol) είναι τα βασικά εργαλεία για την επικοινωνία με το διαδίκτυο. Το πρωτόκολλο TCP / IP, παρέχει την δυνατότητα επικοινωνίας μέσω ενιαίου δικτύου ή διασυνδεδεμένων δικτύων (Internet). Τα παραπάνω πρωτόκολλα μαζί με το SERIAL, IrDA, και το HTTP είναι διαθέσιμα από το LabVIEW στην τοποθεσία Functions → Data Communication → Protocols όπως φαίνεται και στα σχήματα Σχήμα 3.23 και Σχήμα 3.24 . Όπως και τα υπόλοιπα πρωτόκολλα επικοινωνιών έτσι και το TCP πρωτόκολλο επικοινωνίας περιλαμβάνει μια σειρά από διαδικασίες όπως το άνοιγμα της σύνδεσης, η λήψη ή αποστολή δεδομένων, το κλείσιμο της σύνδεσης κλπ.


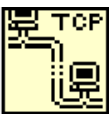





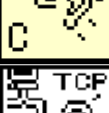
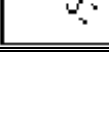



Σχήμα 3.23: protocols

Για την ανάπτυξη τις συγκεκριμένης εφαρμογής γίνεται χρήση του TCP πρωτοκόλλου το οποίο πραγματοποιεί την σύνδεση του arduino με το LabVIEW μέσω του διαδικτύου. Το σύνολο των VIs που διαθέτει το LabVIEW και σχετίζονται με το TCP πρωτόκολλο φαίνονται στο σχήμα 3.24 ενώ η λειτουργία τους παρουσιάζεται σύντομα στον πίνακα 3.2.



Σχήμα 3.24: TCP palette

Εικονίδιο	ονομασία	Επεξήγηση
	TCP Listen.vi	Δημιουργεί έναν listener και περιμένει μια αποδεκτή σύνδεση δικτύου TCP στην προκαθορισμένη θύρα .
	TCP Open Connection	Ανοίγει μια νέα σύνδεση δικτύου TCP με τη διεύθυνση και την απομακρυσμένη θύρα ή το όνομα της υπηρεσίας .
	TCP Read	Διαβάζει έναν αριθμό bytes από μια σύνδεση δικτύου TCP, επιστρέφοντας τα αποτελέσματα στη έξοδο data out.
	TCP Write	Γράφει δεδομένα σε μια TCP σύνδεση δικτύου .
	TCP Close Connection	Κλείνει μια σύνδεση δικτύου TCP .
	IP To String	Μετατρέπει μια IP διεύθυνση δικτύου σε ένα string.
	String To IP	Μετατρέπει ένα string σε μια IP διεύθυνση δικτύου .
	Resolve Machine Alias.vi	Επιστρέφει τη διεύθυνση δικτύου της μηχανής (πχ υπολογιστής).
	TCP Create Listener	Δημιουργεί έναν ακροατή για μια TCP σύνδεση δικτύου .
	TCP Wait On Listener	Περιμένει μια αποδεκτή σύνδεση δικτύου TCP.

Πίνακας 3.2: Περιγραφή λειτουργιών της TCP palette.

Κεφάλαιο 4

Υλοποίηση του συστήματος

4.1 Εισαγωγή

Ο αναλυτής που αναπτύχθηκε στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας έχει την δυνατότητα να υπολογίζει και εμφανίζει στο GUI που αναπτύχθηκε με το LabVIEW την ενεργό τιμή, την μέγιστη τιμή, την ελάχιστη τιμή και την μέση τιμή τόσο της τάσης όσο και του ρεύματος. Επιπλέον υπολογίζει και εμφανίζει την ενεργό ισχύ, την άεργη ισχύ, την φαινόμενη ισχύ, το συντελεστή ισχύος, την κυματομορφή της τάσης, την κυματομορφή του ρεύματος και την συχνότητα της τάσης του δικτύου. Το GUI που αναπτύχθηκε με το LabVIEW έχει σχεδιαστεί έτσι ώστε εκτός από τις στιγμιαίες τιμές να εμφανίζει τις κυματομορφές της ενεργού τιμής της τάσης και του ρεύματος, την κυματομορφή της ενεργού ισχύος και την κυματομορφή της συχνότητας της τάσης του δικτύου. Η επικοινωνία μεταξύ του GUI και του αναλυτή πραγματοποιείται μέσω Ethernet με αποτέλεσμα να είναι εφικτή η λήψη απομακρυσμένων μετρήσεων.

Για τον έλεγχο της κατανάλωσης της ηλεκτρικής ισχύος ο συγκεκριμένος αναλυτής υποστηρίζει 3 επιλογές από τις οποίες ο χρήστης έχει την δυνατότητα να επιλέξει την επιθυμητή. Στην περίπτωση που η μετρούμενη τιμή της ενεργού ισχύος υπερβεί το όριο που έχει επιλέξει ο χρήστης το σύστημα αυτόματα διακόπτει την τροφοδοσία σε ορισμένα μη κρίσιμα φορτία. Το σύστημα που αναπτύχθηκε διαθέτει επίσης μια σειρά ενδεικτικές λυχνίες οι οποίες ενεργοποιούνται σε διάφορες καταστάσεις με σκοπό την ενημέρωση του χρήστη πχ αν η συνολική κατανάλωση υπερβεί τα προκαθορισμένα όρια, αν ο συντελεστής ισχύος είναι κάτω από 0,8 κλπ.

Για την επίτευξη των παραπάνω λειτουργιών αρχικά αναπτύχθηκε με το Arduino IDE το πρόγραμμα που εκτελεί ο μικροελεγκτής ATmega328 της πλατφόρμας Arduino. Το πρόγραμμα αυτό αναλαμβάνει την λήψη των μετρήσεων, την πραγματοποίηση των απαιτούμενων υπολογισμών και την αποστολή τους στο GUI μέσω του Ethernet Shield. Επίσης το πρόγραμμα αυτό υποστηρίζει την εμφάνιση των μετρούμενων μεγεθών στην LCD οθόνη που έχει προστεθεί στην πλατφόρμα Arduino. Για την απομακρυσμένη απεικόνιση των παραπάνω μεγεθών και την απομακρυσμένη διαχείριση του αναλυτή αναπτύχθηκε ένα GUI με το LabVIEW.

4.2 Το πρόγραμμα του ATmega 328

Στην συνέχεια του κεφαλαίου ακολουθεί η ανάλυση του προγράμματος που αναπτύχθηκε και φορτώθηκε στον μικροελεγκτή της πλατφόρμας Arduino (πρόγραμμα 1). Λόγο της πολυπλοκότητας που παρουσιάζει το πρόγραμμα και για την καλύτερη κατανόηση του έχει χωριστεί σε μικρότερα τμήμα που καλούνται «διαδικασίες». Για κάθε διαδικασία αντιστοιχεί μια παράγραφος στην οποία πραγματοποιείται η ανάλυση της.

```
#define samples 200
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Πρόγραμμα 1. Ο κώδικας του Arduino.

```

LiquidCrystal_I2C lcd(0x3F,20,4);
IPAddress ip(192,168,100,101);
IPAddress gateway(192,168,1,254);
IPAddress subnet(255,255,255,0);
EthernetServer server(6234);
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0F, 0xDF, 0x18 };
int new_value = 0;
int old_value = 0;
int pin9 = 9;
int pin6 = 6;
int pin7 = 7;
int pin8 = 8;
int pin5 = 5;
int pin4 = 4;
int pin3 = 3;
int n=5000;
int i = 0;
int f = 0;
int currentlimit=3;
unsigned int counter = 0;
char c, c1;
String buffer ;
String buffer1;
volatile unsigned int samples_f;
volatile boolean adcdone;
volatile boolean temp;
volatile float sum1aa=0.00;
volatile float sum2aa=0.00;
volatile float sum3aa=0.00;
volatile float sum4aa=0.00;
volatile float max1aa = 0.01;
volatile float min1aa = 0.00;
volatile float max2aa = 0.00;
volatile float min2aa = 0.00;
volatile float inst_poweraa = 0;
volatile float sum_inst_poweraa = 0;
volatile int samples0[samples];
volatile int samples1[samples];
float sum1, sum2,sum3,sum4,max1,min1,max2,min2
,mean1,mean2,rms1,rms2,inst_power,sum_inst_power;
float s = 0;
float p = 0;
float q = 0;
float curr;

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).


```

float vol;
float frequency;
float current;
float pf;
float voltage;
void setup()
{
  // Διαδικασία 1
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
  pinMode(pin6, OUTPUT);
  pinMode(pin7, INPUT);
  pinMode(pin8, OUTPUT);
  pinMode(pin9, INPUT);
  lcd.init();
  lcd.backlight();
  Ethernet.begin(mac, ip, gateway, subnet);
  // Διαδικασία 2
  ADMUX = bit(REFS0) | (1 & 0x07); // Επιλογή αναλογικής εισόδου A4
  ADCSRA |= (1 << ADSC); // Ενεργοποίηση λειτουργίας auto trigger του ADC
  ADCSRA |= 1 << ADIFSC; // Ενεργοποίηση της διακοπής του ADC
  ADCSRA |= 1 << ADIFR; // Ενεργοποίηση του ADC
  sei(); // Ενεργοποίηση των διακοπών
  ADCSRA |= 1 << ADSC; // Εκκίνηση της μετατροπής
}
void loop()
{
  // Διαδικασία 1
  digitalWrite(pin5, HIGH); // Ένδειξη λειτουργίας του μικροελεγκτή
  const int button = digitalRead(pin7);
  if (button == LOW) // Έλεγχος γενικής ασφάλειας του κυκλώματος ισχύος
  {
    lcd.setCursor(0, 0);
    lcd.print("MAIN POWER OFF ");
    digitalWrite(pin6, LOW);
  }
  else if (button == HIGH) // Έλεγχος γενικής ασφάλειας του κυκλώματος ισχύος
  {
    digitalWrite(pin6, HIGH); // ενεργοποίηση της ένδειξης σχετικά με την
//κατάσταση της γενικής ασφάλειας.
  }
  // Διαδικασία 2
  EthernetClient client = server.available(); // Έλεγχος εάν υπάρχει συνδεδεμένος χρήστης
  if (client)

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" REMOTE CONTROLL ");
  lcd.setCursor(5, 1);
  lcd.print(" IS ON ");
  lcd.setCursor(6, 2);
  lcd.print(" T.E.I. ");
  lcd.setCursor(2, 3);
  lcd.print(" DITIKHS ELLADOS ");
  delay(100);
  while (client.connected())
  {
    while (client.available()>0)
    {
      p=sum_inst_poweraa / samples; // Υπολογισμός της πραγματικής ισχύος
      rms1 =sqrt(sum2aa/samples); // Υπολογισμός της RMS τιμής της τάσης
      rms2 =sqrt(sum4aa/samples); // Υπολογισμός της RMS τιμής του ρεύματος
      s=rms1 * rms2; // Υπολογισμός της φαινόμενης ισχύος
      pf= p/s; // Υπολογισμός του συντελεστή ισχύος
      if (pf<0.8)
      {
        digitalWrite(pin3,HIGH);
      }
      else
      {
        digitalWrite(pin3,LOW);
      }
      int u = digitalRead(pin9);
      if ((s < n )&& (u==HIGH))
      {
        digitalWrite(pin4,LOW);
      }
      if (s > n)
      {
        digitalWrite(pin4,HIGH);
      }
      // Διαδικασία 3
      c = client.read(); //Ανάγνωση μηνύματος που έχει αποστείλει από το LabVIEW
      buffer += c;
    }
    if (buffer == "limit1")
    {
      n =8000;
    }
  }
}

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

client.println("1");
buffer = "";
}

else if (buffer == "power_limit2")
{
n = 10000;
client.println("2");
buffer = "";
}

else if (buffer == "power_limit3")
{
n = 15000;
client.println("3");
buffer = "";
}
// Διαδικασία 4
else if (buffer == "frequency")
{
if (adcdone==true)
{
frequency=(1/(samples_f*0.21)*1000);//Υπολογισμός της συχνότητας του δικτύου
client.println(samples_f);
client.println(frequency);
adcdone=false;
}
buffer = "";
}
// Διαδικασία 5
else if (buffer == "rms")
{
rms1=sqrt(sum2aa/samples); //Υπολογισμός της RMS τιμής της τάσης
rms2 = sqrt(sum4aa/samples); //Υπολογισμός της RMS τιμής του ρεύματος
client.println(rms2);
client.println(rms1);
buffer = "";
}
// Διαδικασία 6
else if (buffer == "mean")
{
mean2= sum3aa/samples; //Υπολογισμός της μέσης τιμής της τάσης
mean1= sum1aa/samples; //Υπολογισμός της μέσης τιμής του ρεύματος
client.println(mean2);

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

client.println(mean1);
buffer = "";
}
// Διαδικασία 7
else if (buffer == "max-min")
{
client.println(max2aa);
client.println(min2aa);
client.println(max1aa);
client.println(min1aa);
buffer = "";
}
// Διαδικασία 8
else if (buffer == "power")
{
rms1 = sqrt(sum2aa/samples);
rms2 = sqrt(sum4aa/samples);
p=sum_inst_poweraa / samples;
s=rms1 * rms2;
q=sqrt((s*s)-(p*p));
client.println(p);
client.println(s);
client.println(q);
s=0;
p=0;
q=0;
buffer = "";
}
// Διαδικασία 9
else if (buffer == "all")
{
p=sum_inst_poweraa / samples;
mean1= sum1aa/samples;
rms1 = sqrt(sum2aa/samples);
mean2= sum3aa/samples;
rms2 = sqrt(sum4aa/samples);
s=rms1 * rms2;
client.println(mean1);
client.println(rms1);
client.println(max1aa);
client.println(min1aa);
client.println(mean2);
client.println(rms2);
client.println(max2aa);

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

client.println(min2aa);
client.println(s);
client.println(p);
s=0;
p=0;
buffer = "";
}
// Διαδικασία 10
else if (buffer == "waveform")
{
if (temp==true)
{
cli();
for (int j=0 ; j < samples ;j++)
{
curr=(((samples0[j] -510)*75.757/1023)*20);
vol=((samples1[j] -512)*0.7475);
client.println(curr);
client.println(vol);
}
temp=false;
buffer = "";
}
sei();
}
}
lcd.clear();
}
// Διαδικασία 11
else
{
rms1 = sqrt(sum2aa/samples);
rms2 = sqrt(sum4aa/samples);
p=sum_inst_poweraa / samples;
s=rms1 * rms2;
pf= ((p/s)-0.001);
if (adcdone==true)
{
frequency=(1/(samples_f *0.21)*1000);
adcdone=false;
}
if (pf<0.8)
{
digitalWrite(pin3,HIGH);
}
}
}

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

}
else
{
    digitalWrite(pin3,LOW);
}
int u = digitalRead(pin9);

if ((s < n) && (u==HIGH))
{
    digitalWrite(pin4,LOW);
}
if (s >= n)
{
    digitalWrite(pin4,HIGH);
}
lcd.setCursor(0, 0);
lcd.print("V=");
lcd.print(rms2);
lcd.setCursor(8, 0);
lcd.print("V");
lcd.setCursor(0, 1);
lcd.print("I=");
lcd.print(rms1);
lcd.setCursor(8, 1);
lcd.print(" A ");
lcd.setCursor(0, 2);
lcd.print("PF=");
lcd.setCursor(3, 2);
lcd.print(pf);
lcd.setCursor(0, 3);
lcd.print("F=");
lcd.print(frequency);
lcd.setCursor(7, 3);
lcd.print("Hz");
lcd.setCursor(10, 0);
lcd.print("P=");
lcd.print(p);
lcd.setCursor(19, 0);
lcd.print("W");
lcd.setCursor(10, 1);
lcd.print("S=");
lcd.print(s);
lcd.setCursor(18, 1);
lcd.print("VA");

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

    lcd.setCursor(10, 2);
    lcd.print(" TEI DYT.");
    lcd.setCursor(10, 3);
    lcd.print("ELLADOS");
    rms1=0;
    rms2=0;
    delay(500);
  }
}
}

```

```

ISR(ADC_vect)           // Η ρουτίνα εξυπηρέτησης διακοπής του ADC
{
// Διαδικασία 1
uint8_t theLow = ADCL;           // Αποθήκευση του LSB του αποτελέσματος
    //της μετατροπής
uint16_t theTenBitResult=ADCH << 8|theLow; // Αποθήκευση του αποτελέσματος της
    //μετατροπής
switch (ADMUX)                 // Επιλογή αναλογικής εισόδου
{
  case 0x41:                    // Όταν η αναλογική είσοδος είναι η A1 τότε:
    samples0[i] = theTenBitResult; // Αποθήκευση του αποτελέσματος του ADC
    // στον πίνακα samples0[i]
    ADMUX = 0x43;              // Αλλαγή του αναλογικού εισόδου σε A3 αντί για A1
    i = i + 1;                 // Αύξηση του μετρητή "i"
    break;
  case 0x43:                    // Όταν η αναλογική είσοδος είναι η A3 τότε:
    samples1[f] = theTenBitResult; // Αποθήκευση του αποτελέσματος του ADC
    // στον πίνακα samples1[f]
    ADMUX = 0x41;              // Καθορισμός αναλογικού εισόδου A1 αντί για A3
    f = f + 1;                 // Αύξηση του μετρητή "f"
    break;
  default:
    break;
}
// Διαδικασία 2
if (f < samples)
{
  if (i < samples)
  {
    if (i==f)
    {
      old_value = new_value;

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).

```

new_value = theTenBitResult;
if (old_value < 512 && new_value >=512) //Εντοπισμός του σημείου μηδενισμού
{
  samples_f = counter;
  counter = 0;
  adcdone=true;
}
else
{
  counter = counter +1;
}
current=((samples0[f] -510)*75.757/1023); // Στιγμαία τιμή του ρεύματος
voltage=((samples1[i] -512)*0.039*19.1667); // Στιγμαία τιμή της τάσης
inst_power=voltage*current; //Υπολογισμός της στιγμιαίας ισχύος
sum_inst_power = sum_inst_power + inst_power;
if (current > max1)
{
  max1=current;
}
if (current < min1)
{
  min1=current;
}
if (voltage > max2)
{
  max2=voltage;
}
if (voltage < min2)
{
  min2=voltage;
}
sum1 = sum1+(current);
sum2=sum2 + (current * current);
sum3 = sum3+(voltage);
sum4=sum4 + (voltage * voltage);
}
}
}
// Διαδικασία 3
else
{
  sum1aa=sum1;
  sum2aa=sum2;
  sum3aa=sum3;

```

Πρόγραμμα 1. Ο κώδικας του Arduino (Συνέχεια).


```

sum4aa=sum4;
min1aa=min1;
max1aa=max1;
min2aa=min2;
max2aa=max2;
inst_poweraa=inst_power;
sum_inst_poweraa=sum_inst_power;
sum1=0;
sum2=0;
sum3=0;
sum4=0;
min1=0;
max1=0;
min2=0;
max2=0;
inst_power=0;
sum_inst_power=0;
i=0;
f=0;
temp=true;
}
}

```

Πρόγραμμα 1. Ο κώδικας του Arduino

Δήλωση των γενικών μεταβλητών

Αρχικά στο πρόγραμμα που φορτώθηκε στο Arduino γίνεται η δήλωση των σταθερών, των γενικών μεταβλητών και των συμπεριλαμβανομένων βιβλιοθηκών. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι η SPI, η Ethernet, η Wire και η LiquidCrystal_I2C. Οι βιβλιοθήκες SPI και Ethernet απαιτούνται για την λειτουργία του Ethernet Shield το οποίο είναι συνδεδεμένο με το Arduino. Ενώ οι βιβλιοθήκες Wire και LiquidCrystal_I2C αξιοποιούνται για την επικοινωνία του μικροελεγκτή με την LCD οθόνη. Στη συνέχεια ακολουθεί η δήλωση των γενικών μεταβλητών που χρησιμοποιούν στο πρόγραμμα.

Η συνάρτηση setup

Η συνάρτηση setup έχει χωριστεί σε δυο διαδικασίες, στην πρώτη πραγματοποιείται ο ορισμός των ψηφιακών ακροδεκτών ως είσοδοι ή έξοδοι. Ποιο συγκεκριμένα οι ακροδέκτες 3,4,5,6,8 προγραμματίζονται να λειτουργούν ως έξοδοι μέσω της συνάρτησης pinMode(pin, OUTPUT). Ενώ οι ακροδέκτες 7 και 9 προγραμματίζονται να λειτουργούν ως είσοδοι μέσω της συνάρτησης pinMode(pin, INPUT). Στην συνέχεια πραγματοποιούνται οι αρχικοποιήσεις της LCD οθόνης ,του Ethernet Shield και τέλος γίνεται η ενεργοποίηση του ADC. Η ενεργοποίηση του ADC πραγματοποιείται δίνοντας κατάλληλες τιμές στους καταχωρητές ADMUX, ADCSRA. Με τον τρόπο αυτό επιτυγχάνεται η ενεργοποίηση της αυτόματης λειτουργίας του ADC, η ενεργοποίηση των διακοπών, η ενεργοποίηση της διακοπής που προκαλεί ο ADC κατά την ολοκλήρωση μιας μετατροπής και τέλος τίθεται σε λειτουργία ο ADC.

Η συνάρτηση loop

Στην συνάρτηση loop έχει γραφτεί το μέρος του προγράμματος το οποίο εκτελείται επανειλημμένα από τον μικροελεγκτή. Λόγο της πολυπλοκότητας που παρουσιάζει έχει χωριστεί σε 11 διαδικασίες οι οποίες αναλύονται στην συνέχεια.

Στην πρώτη διαδικασία ενεργοποιείται ένα LED ώστε να ενημερωθεί ο χρήστης ότι ο μικροελεγκτής ξεκίνησε την εκτέλεση του προγράμματος. Στην συνέχεια πραγματοποιείται ο έλεγχος της κατάστασης της γενικής ασφάλειας ισχύος. Αν η ασφάλεια έχει διακόψει το κύκλωμα ισχύος λόγω υπερέντασης ή βραχυκυκλώματος, τότε απενεργοποιείται το αντίστοιχο LED και τυπώνεται στην ενσωματωμένη οθόνη το μήνυμα "MAIN POWER IS OFF". Στην περίπτωση που η γενική ασφάλεια δεν έχει πρόβλημα, ο μικροελεγκτής ελέγχει εάν έχει συνδεθεί κάποιος απομακρυσμένος χρήστης (διαδικασία 2). Εάν υπάρχει συνδεδεμένος χρήστης ο μικροελεγκτής θα εκτελέσει το μέρος του προγράμματος που αντιστοιχεί στον απομακρυσμένο έλεγχο (διαδικασίες 2-10), αλλιώς θα εκτελέσει το μέρος του προγράμματος που αντιστοιχεί στην τοπική λειτουργία (διαδικασία 11).

Στην διαδικασία 2 της συνάρτησης loop αρχικά τυπώνεται στην οθόνη το μήνυμα "REMOTE CONTROL IS ON" και παραμένει μέχρι ο χρήστης να αποσυνδεθεί. Ο μικροελεγκτής πραγματοποιεί τους υπολογισμούς των ηλεκτρικών μεγεθών και περιμένει ένα μήνυμα από τον συνδεδεμένο χρήστη με σκοπό την αποστολή των αντίστοιχων τιμών. Στην συγκεκριμένη διαδικασία επιπλέον γίνεται έλεγχος αν ο συντελεστής ισχύος είναι μικρότερος από 0.8.

Στη διαδικασία 3 έχει γραφτεί το κομμάτι του προγράμματος το οποίο δίνει την δυνατότητα στο χρήστη να επιλέξει ένα από τα προκαθορισμένα όρια της καταναλισκόμενης ισχύος. Οι διαθέσιμες επιλογές είναι τρεις (8 KVA, 10 KVA και 15 KVA), και η επιλογή τους πραγματοποιείται μέσω του LabVIEW. Όταν καμία από τις τρεις επιλογές δεν έχει επιλεγεί τότε το όριο που έχει οριστεί για την καταναλισκόμενη ισχύ είναι τα 5 KVA. Εάν η ισχύς που καταναλώνεται υπερβεί το επιλεγμένο όριο διακόπτεται μέσω ενός ρελέ η τροφοδοσία των μη κρίσιμων φορτίων.

Στην διαδικασία 4 πραγματοποιείται ο υπολογισμός της συχνότητας της τάσης του δικτύου αξιοποιώντας τα δεδομένα που συλλέχτηκαν κατά την εκτέλεση της ρουτίνας εξυπηρέτησης της διακοπής του ADC. Για τον υπολογισμό της συχνότητας λαμβάνεται υπόψη ο χρόνος μεταξύ δύο διαδοχικών δειγμάτων ο οποίος καθορίζεται από την επιλεγμένη συχνότητα δειγματοληψίας που είναι 4,8KHz. Επομένως ο παραπάνω χρόνος προσδιορίζεται σε $1/4800=208\mu\text{s}$. Η περίοδος του σήματος υπολογίζεται πολλαπλασιάζοντας τον αριθμό των δειγμάτων που αντιστοιχούν σε μια πλήρη περίοδο του σήματος, με το χρόνο που μεσολαβεί μεταξύ δύο διαδοχικών δειγμάτων. Τέλος αφού υπολογίστηκε η περίοδος του σήματος τότε αξιοποιώντας τον τύπο $F=1/T$ πραγματοποιείται ο υπολογισμός της συχνότητας. Η συγκεκριμένη διαδικασία εκτελείται εφόσον ο μικροελεγκτής λάβει μέσω Ethernet την εντολή "frequency".

Στην διαδικασία 5 πραγματοποιείται ο υπολογισμός και η αποστολή της ενεργούς τιμής της τάσης και της ενεργούς τιμής του ρεύματος. Για τον υπολογισμό της ενεργούς τιμής πραγματοποιείται η διαίρεση του αθροίσματος των τετραγώνων των δειγμάτων του αντίστοιχου μεγέθους με τον συνολικό αριθμό τους. Το συγκεκριμένο τμήμα του προγράμματος εκτελείται όταν ο χρήστης επιλέξει να παρακολουθήσει την ενεργό τιμή τάσης και ρεύματος μέσω του GUI. Σε αυτή την περίπτωση το GUI αποστέλλει την εντολή "rms".

Στην διαδικασία 6 που εκτελείται όταν ληφθεί η εντολή "mean", πραγματοποιείται ο υπολογισμός και η αποστολή των μέσων τιμών της τάσης και του ρεύματος. Η μέση τιμή

υπολογίζεται διαιρώντας το άθροισμα των τιμών των δειγμάτων του αντίστοιχου μεγέθους με τον συνολικό αριθμό των δειγμάτων. Ενώ η διαδικασία 7 που εκτελείται όταν ληφθεί η εντολή "max-min" και είναι υπεύθυνη για τον προσδιορισμό της μέγιστης και της ελάχιστης τιμής τόσο της τάσης όσο και του ρεύματος.

Ο υπολογισμός της ισχύος που καταναλώνεται πραγματοποιείται στην διαδικασία 8 μετά την λήψη της εντολής "power". Αρχικά υπολογίζεται η ενεργός τιμή της τάσης και του ρεύματος οι οποίες απαιτούνται για τον υπολογισμό της φαινόμενης ισχύος. Οι τιμές που υπολογίζονται στην συγκεκριμένη διαδικασία είναι η τιμή της ενεργού ισχύος, η τιμή της φαινόμενης ισχύος και τέλος υπολογίζεται η τιμή της άεργου ισχύος. Για τον υπολογισμό της ενεργού ισχύος πραγματοποιείται η διαίρεση του αθροίσματος των τιμών της στιγμιαίας ισχύος που υπολογίζονται σε μια περίοδο με τον συνολικό αριθμό τους. Ενώ ο υπολογισμός της φαινόμενης ισχύος πραγματοποιείται πολλαπλασιάζοντας την RMS τιμή της τάσης με την RMS τιμή του ρεύματος. Τέλος η τιμή της άεργου ισχύος προκύπτει από τον υπολογισμό της τετραγωνικής ρίζας της διαφοράς της φαινόμενης ισχύος στο τετράγωνο με την τιμή της ενεργού ισχύος στο τετράγωνο.

Η διαδικασία 9 εκτελείται μετά την λήψη της εντολής "all" και υλοποιεί όλους του ς παραπάνω υπολογισμούς ώστε να αποστέλλονται ταυτόχρονα στο GUI όλα τα ηλεκτρικά μεγέθη που υπολογίστηκαν. Σε αυτή την περίπτωση τα παραπάνω ηλεκτρικά μεγέθη εμφανίζονται στο GUI ταυτόχρονα.

Η κυματομορφή ενός σήματος είναι πολύ σημαντική αφού από αυτή μπορούν εξαχθούν χρήσιμα συμπεράσματα. Για τον σκοπό αυτό ο συγκεκριμένος αναλυτής ενέργειας έχει την δυνατότητα να αποστέλλει στο GUI δύο πίνακες των 200 στοιχείων που περιέχουν τις στιγμιαίες τιμές της τάσης και του ρεύματος ώστε να είναι εφικτή η αναπαράσταση των κυματομορφών της τάσης του δικτύου και του ρεύματος του φορτίου στο GUI. Η αποστολή των παραπάνω δεδομένων πραγματοποιείται στην διαδικασία 10 η οποία εκτελείται όταν ληφθεί η εντολή "waveform".

Το υπόλοιπο μέρος του κώδικα αποτελεί την διαδικασία 11 η οποία εκτελείται όταν δεν υπάρχει απομακρυσμένος χρήστης (τοπική λειτουργία). Κατά την εκτέλεση αυτής της διαδικασίας πραγματοποιούνται οι υπολογισμοί των παρακάτω ηλεκτρικών μεγεθών:

- Ενεργός τιμή της τάσης
- Ενεργός τιμή του ρεύματος
- Ενεργό ισχύ
- Φαινόμενη ισχύ
- Συντελεστή ισχύος
- Συχνότητα της τάσης του δικτύου

Οι τιμές των ηλεκτρικών μεγεθών που υπολογίζονται εμφανίζονται στην οθόνη που έχει ενσωματωθεί στην πλατφόρμα Arduino ώστε να είναι διαθέσιμες στον τοπικό χρήστη. Επίσης και στην τοπική λειτουργία του συστήματος πραγματοποιείται ο έλεγχος της καταναλισκόμενης ισχύος σύμφωνα είτε με το προκαθορισμένο όριο των 5KVA είτε με το όριο που είχε ορίσει ο χρήστης κατά την απομακρυσμένη λειτουργία.

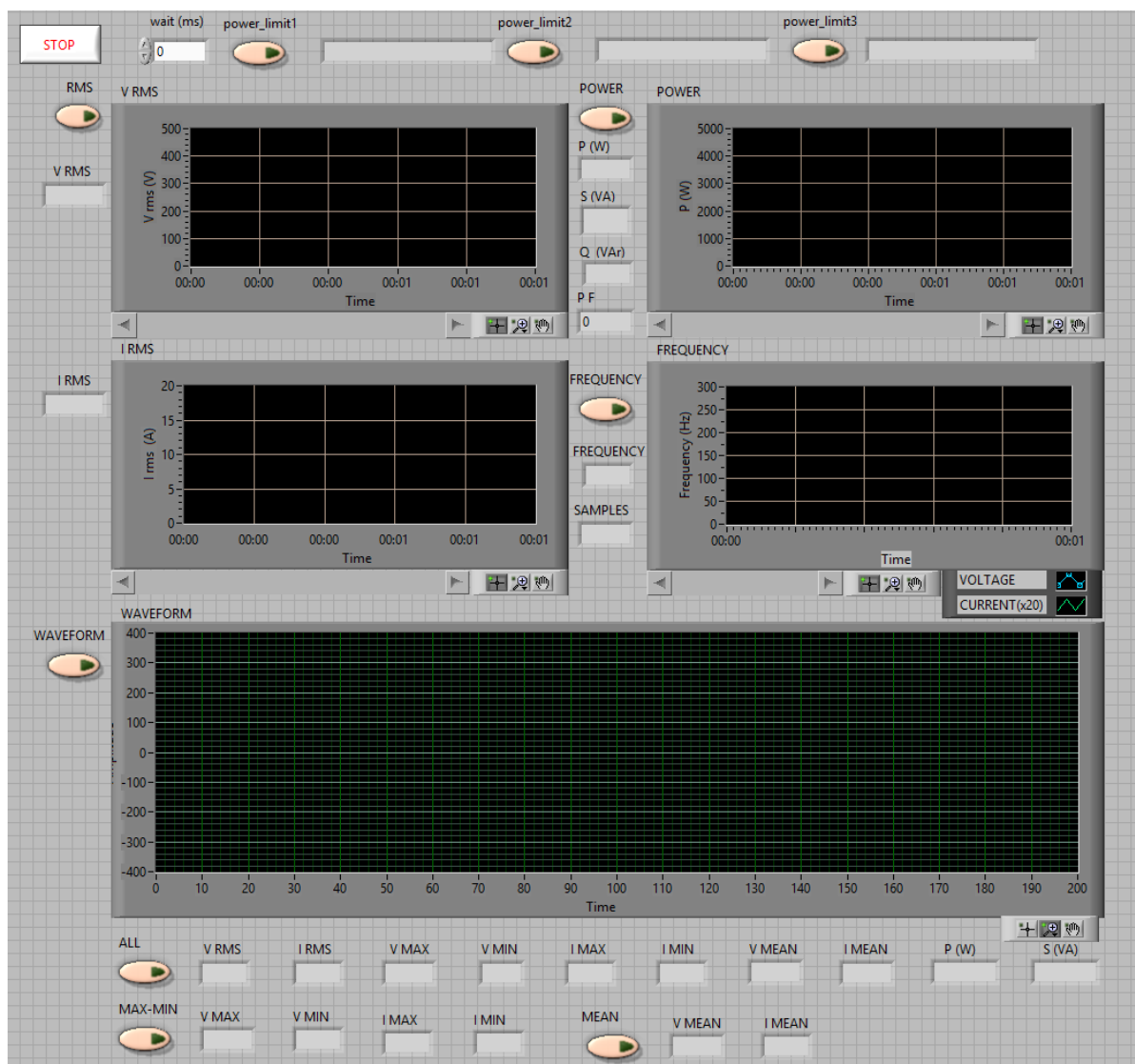
Η ρουτίνα εξυπηρέτησης της διακοπής του ADC

Η ρουτίνα εξυπηρέτησης της διακοπής του ADC αποτελεί το πλέον σημαντικό τμήμα του προγράμματος. Η ρουτίνα αυτή είναι υπεύθυνη για την συλλογή των μετρήσεων και εκτελείται κάθε φορά που ο ADC ολοκληρώνει μια μετατροπή. Ο ATmega328 διαθέτει 6 αναλογικές εισόδους οι οποίες καταλήγουν στον ADC μέσω ενός πολυπλέκτη. Συνεπώς ο

ATmega328 δεν μπορεί να πραγματοποιήσει την ταυτόχρονη δειγματοληψία σε περισσότερες από μια αναλογικές εισόδους. Επειδή στην συγκεκριμένη εφαρμογή θα πρέπει να λαμβάνονται δείγματα από δύο αναλογικές εισόδους κατά την εκτέλεση της ρουτίνας εξυπηρέτησης διακοπής θα πρέπει να επιλέγεται η αναλογική είσοδος που θα χρησιμοποιηθεί από τον ADC στην επόμενη μετατροπή. Για τον σκοπό αυτό στην διαδικασία 1 αρχικά αποθηκεύεται προσωρινά το αποτέλεσμα της μετατροπής του ADC. Έπειτα ανάλογα με την επιλεγμένη αναλογική είσοδο το αποτέλεσμα της μετατροπής αποθηκεύεται στον πίνακα `samples0` ή στον πίνακα `samples1` και μέσω του καταχωρητή `ADMUX` καθορίζεται η αναλογική είσοδος που θα χρησιμοποιηθεί στην επόμενη μετατροπή του ADC. Το γεγονός αυτό έχει ως αποτέλεσμα να λαμβάνονται διαδοχικά μια τιμή της τάσης και μια τιμή του ρεύματος με την μισή συχνότητα δειγματοληψίας. Παρόλο που ο ADC έχει ρυθμιστεί να λειτουργεί με συχνότητα δειγματοληψίας 9,6kHz η συχνότητα δειγματοληψίας των δυο μετρούμενων μεγεθών είναι 4,8 kHz.

Για τον υπολογισμό κάποιων ηλεκτρικών μεγεθών πρέπει να πραγματοποιηθεί δειγματοληψία σε τουλάχιστον μια ολοκληρωμένη περίοδο. Λαμβάνοντας υπόψη την συχνότητα δειγματοληψίας και την συχνότητα του δικτύου, έχει οριστεί ένα όριο των διακοσίων δειγμάτων για το κάθε σήμα το οποίο αντιστοιχεί σε περισσότερες από δυο περιόδους για την συχνότητα των 50 Hz. Το γεγονός αυτό επιτρέπει τον συγκεκριμένο αναλυτή να λειτουργεί και σε συχνότητες μικρότερες των 50 Hz. Στην Διαδικασία 2 της ρουτίνας εξυπηρέτησης της διακοπής του ADC πραγματοποιείται ο έλεγχος του αριθμού των δειγμάτων που έχει λάβει ο μικροελεγκτής για το κάθε σήμα. Όταν ο αριθμός αυτός είναι μικρότερος από διακόσια τότε επαναλαμβάνεται η διαδικασία λήψης νέου δείγματος μέχρι να συμπληρωθεί το προκαθορισμένο όριο. Επιπλέον εκτελείται η διαδικασία εντοπισμού του μηδενικού σημείου και ο υπολογισμός του αριθμού των δειγμάτων που λαμβάνονται σε μια περίοδο. Η παραπάνω διαδικασία απαιτείται για τον υπολογισμό της συχνότητας ο οποίος πραγματοποιείται στην συνάρτηση `loop`. Στην συνέχεια πραγματοποιείται ο υπολογισμός της στιγμιαίας τιμής της τάσης και του ρεύματος. Επίσης υπολογίζεται η τιμή της στιγμιαίας ισχύος και το άθροισμα των τιμών της στιγμιαίας ισχύος. Επιπλέον εάν η τιμή του δείγματος είναι μεγαλύτερη ή μικρότερη σε σχέση με τα προηγούμενα δείγματα τότε ενημερώνεται η καταχωρημένη τιμή της μέγιστης ή της ελάχιστης τιμής αντίστοιχα. Τέλος πραγματοποιείται ο υπολογισμός των αθροισμάτων των στιγμιαίων τιμών της τάσης και του ρεύματος, καθώς και των αθροισμάτων των τετραγώνων των στιγμιαίων τιμών της τάσης και του ρεύματος.

Όταν τα δύο πακέτα των διακοσίων δειγμάτων που περιέχουν τις στιγμιαίες τιμές των τάσεων και των ρευμάτων συμπληρωθούν όλες τα μεγέθη που υπολογίστηκαν αντιγράφονται σε νέες μεταβλητές (Διαδικασία 3) με σκοπό να χρησιμοποιηθούν στην συνάρτηση `loop`. Στην συνέχεια οι μεταβλητές που περιείχαν τους αρχικούς υπολογισμούς μηδενίζονται ώστε να ξεκινήσει η συλλογή των νέων δειγμάτων.



Σχήμα 4.1: Το Front Panel της εφαρμογής.

4.3 Το LabVIEW interface

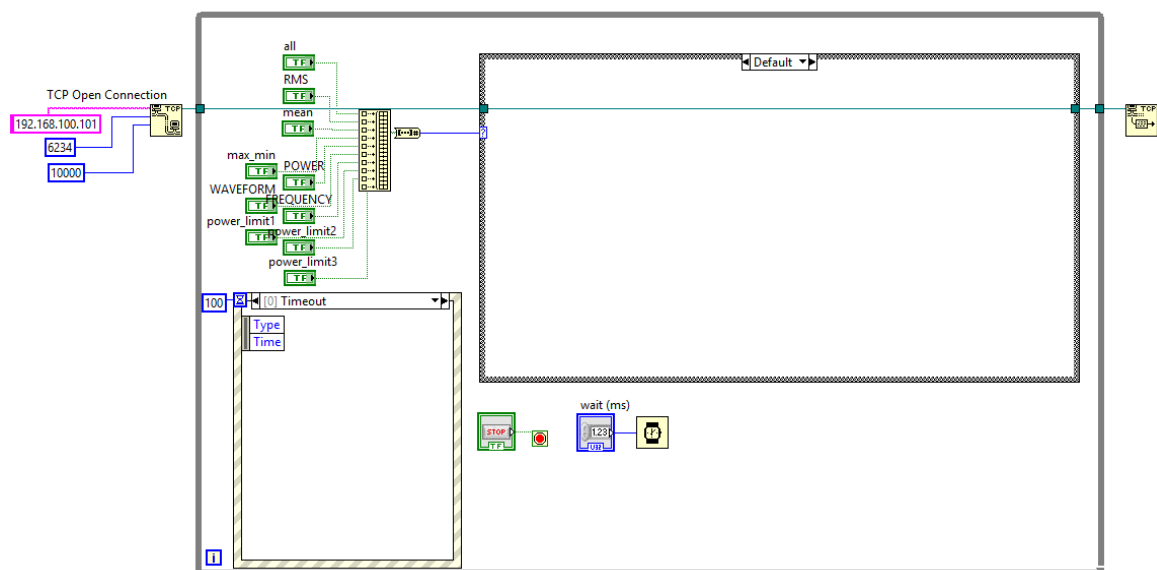
Το λογισμικό πακέτο LabVIEW διαθέτει ένα πλούσιο γραφικό περιβάλλον το οποίο αξιοποιήθηκε για την ανάπτυξη του GUI που έχει σκοπό την παρουσίαση των ηλεκτρικών μεγεθών που υπολογίζονται από τον μικροελεγκτή ATmega 328. Η παρουσίαση των παραπάνω τιμών πραγματοποιείται σε αριθμητικούς ενδείκτες και γραφικές παραστάσεις. Ένα σημαντικό χαρακτηριστικό της εφαρμογής που αναπτύχθηκε είναι ότι η λήψη των δεδομένων που παρουσιάζονται πραγματοποιείται μέσω του διαδικτύου, δηλαδή η επικοινωνία της εφαρμογής με την πλατφόρμα Arduino πραγματοποιείται αξιοποιώντας το Ethernet Shield. Τέλος εκτός από την παρουσίαση των ηλεκτρικών μεγεθών που αποστέλλονται από τον μικροελεγκτή, το GUI διαθέτει μια επιπλέον λειτουργία η οποία επιτρέπει στον χρήστη να επιλέξει ένα από τα τρία διαθέσιμα όρια της καταναλισκομένης ισχύος.

4.3.1 Το Front Panel της εφαρμογής.

Στο σχήμα 4.1 παρουσιάζεται το Front Panel του GUI όπως τελικά διαμορφώθηκε. Το γραφικό περιβάλλον του Front Panel αποτελείται ένα σύνολο από Controls και Indicators που αναλαμβάνουν το έλεγχο του αναλυτή ενέργειας και την εμφάνιση υπολογιζόμενων ηλεκτρικών μεγεθών.

4.3.2 Το Block Diagram της εφαρμογής

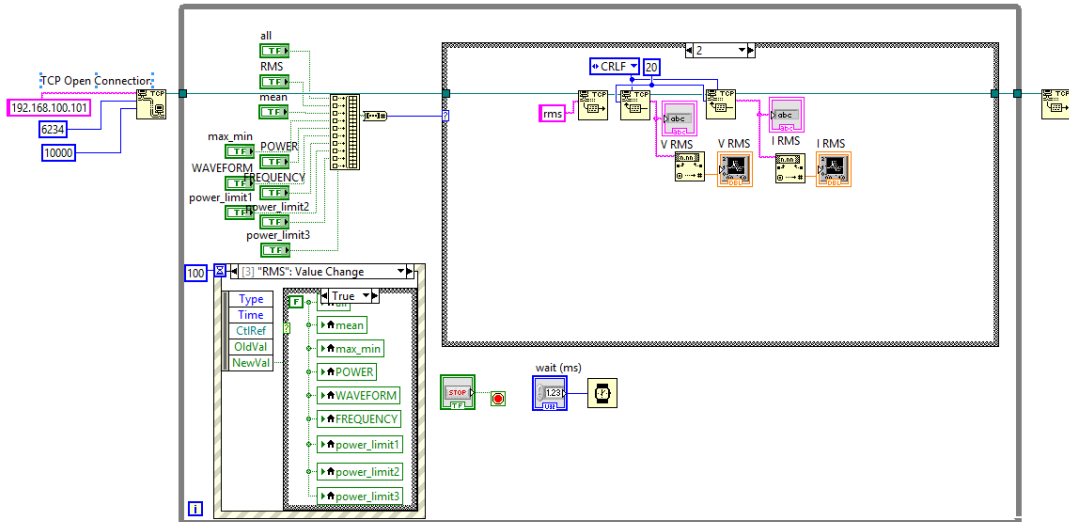
Πίσω από το γραφικό περιβάλλον του Front Panel, βρίσκεται το Block Diagram όπου και είναι διατυπωμένος ο κώδικας του VI (σχήμα 4.2) με χρήση γλώσσας προγραμματισμού G. Μέσω του κώδικα αυτού γίνεται όλη η διαδικασία για την απομακρυσμένη επικοινωνία, την λήψη και απεικόνιση των δεδομένων που αποστέλλονται από τον μικροελεγκτή.



Σχήμα 4.2: Το Block Diagram του VI

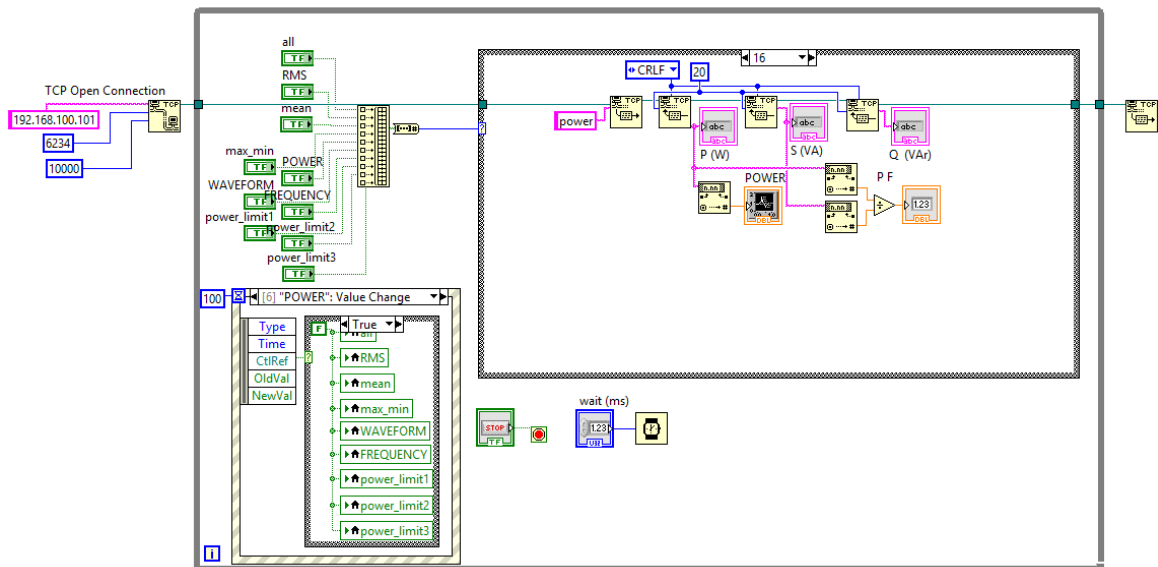
Ο κώδικας του Block Diagram αποτελείται από κάποια συγκεκριμένα μέρη τα οποία είναι υπεύθυνα για την σύνδεση με τον μικροελεγκτή μέσω διαδικτύου, την λήψη δεδομένων και την απεικόνιση τους. Αρχικά χρησιμοποιώντας την συνάρτηση TCP Open Connection πραγματοποιείται η σύνδεση του GUI με τον μικροελεγκτή μέσω του Ethernet Shield. Στην συνέχεια μέσω της Event Case η οποία εκτελείται κάθε 100ms ελέγχεται αν ο χρήστης επέλεξε κάποιο από τα 10 διαθέσιμα κουμπιά του Front Panel. Κάθε φορά που ο χρήστης ενεργοποιεί κάποιο κουμπί τότε μέσω της Event Case απενεργοποιούνται όλα τα υπόλοιπα κουμπιά. Στην συνέχεια εκτελείται η περίπτωση της δομής Case που αντιστοιχεί στο επιλεγμένο κουμπί ώστε να αποσταλεί στον μικροελεγκτή το αντίστοιχο μήνυμα. Έπειτα γίνεται η ανάγνωση και η εμφάνιση των δεδομένων που στέλνει ο μικροελεγκτής.

Αναλυτικότερα όταν επιλεγθεί το κουμπί “RMS” τότε εκτελείται το μέρος του κώδικα που παρουσιάζεται στο σχήμα 4.3. Σε αυτή την περίπτωση το μήνυμα που στέλνεται στον μικροελεγκτή είναι το “rms”, ενώ τα αποτελέσματα που επιστρέφει ο μικροελεγκτής είναι οι ενεργές τιμές της τάσης και του ρεύματος. Επειδή ο μικροελεγκτής στέλνει τις τιμές σε μορφή string, αρχικά πραγματοποιείται η απεικόνιση τους και η μετατροπή τους από string σε αριθμητικές τιμές ώστε να είναι εφικτή η γραφική τους αναπαράσταση παράσταση.



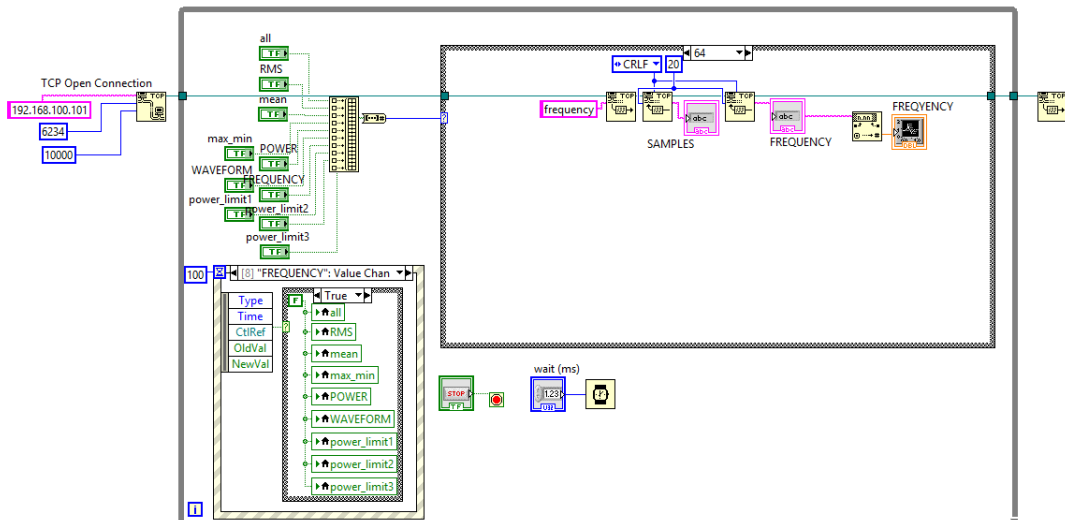
Σχήμα 4.3: Η περίπτωση RMS

Όταν επιλεχτεί το κουμπί “POWER” εκτελείται ο κώδικας του σχήματος 4.4. Στην συγκεκριμένη περίπτωση ο μικροελεγκτής επιστρέφει την τιμή της ενεργού ισχύος, την τιμή της φαινόμενης ισχύος και την τιμή της άεργου ισχύος. Επίσης πραγματοποιείται και ο υπολογισμός του συντελεστή ισχύος.



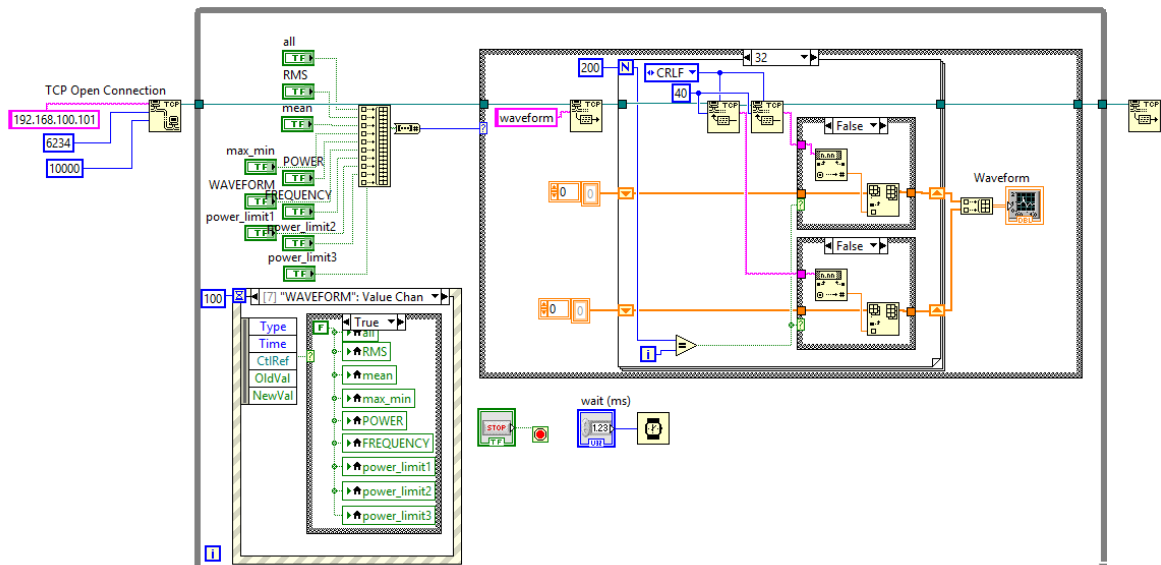
Σχήμα 4.4: Η περίπτωση POWER

Στο σχήμα 4.5 παρουσιάζεται ο κώδικας που εκτελείται όταν επιλεχτεί το κουμπί “FREQUENCY”. Οι τιμές που στέλνει ο μικροελεγκτής σε αυτή την περίπτωση είναι δυο, ο αριθμός των δειγμάτων που συλλέχθηκαν σε μια περίοδο του σήματος και η τιμή της συχνότητας της τάσης του δικτύου. Η τιμή της συχνότητας επειδή απεικονίζεται και σε γραφική παράσταση, μετατρέπεται από string σε αριθμητική τιμή.



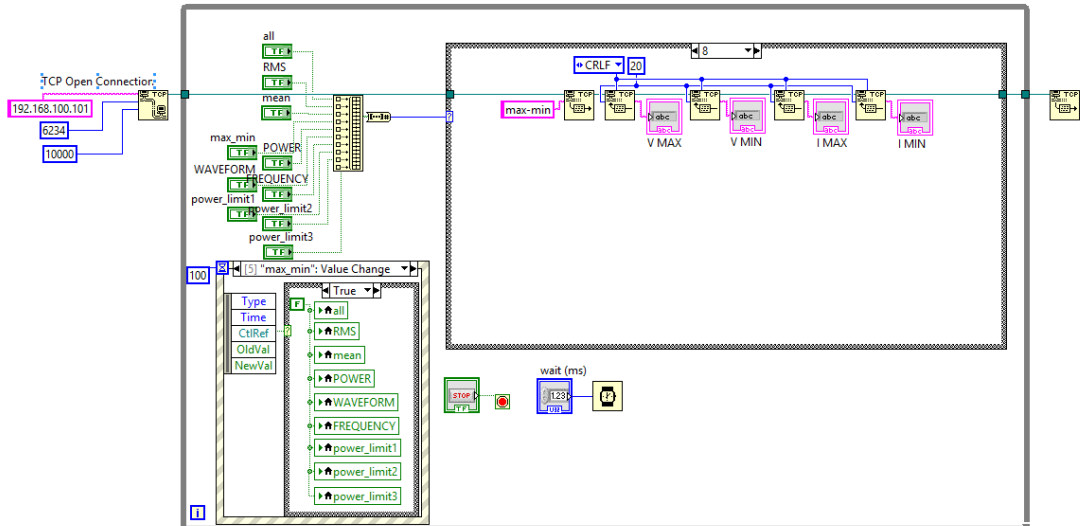
Σχήμα 4.5: Η περίπτωση FREQUENCY

Η απεικόνιση της κυματομορφής της τάσης του δικτύου και του ρεύματος του φορτίου πραγματοποιείται από τον κώδικα του σχήματος 4.6 που εκτελείται όταν επιλεγεί το κουμπί WAVEFORM. Σε αυτή την περίπτωση πραγματοποιείται η λήψη των τιμών που αποστέλλονται από τον μικροελεγκτή και η απεικόνισή τους ως κυματομορφές. Οι τιμές που λαμβάνονται αφού μετατραπούν από string σε αριθμητικές τιμές αποθηκεύονται σε δύο μονοδιάστατους πίνακες. Στον ένα πίνακα αποθηκεύονται οι στιγμιαίες τιμές της τάσης ενώ στον δεύτερο πίνακα αποθηκεύονται οι στιγμιαίες τιμές του ρεύματος. Και οι δύο πίνακες αναπαρίστανται ως γραφικές παραστάσεις στο indicator "Waveform".



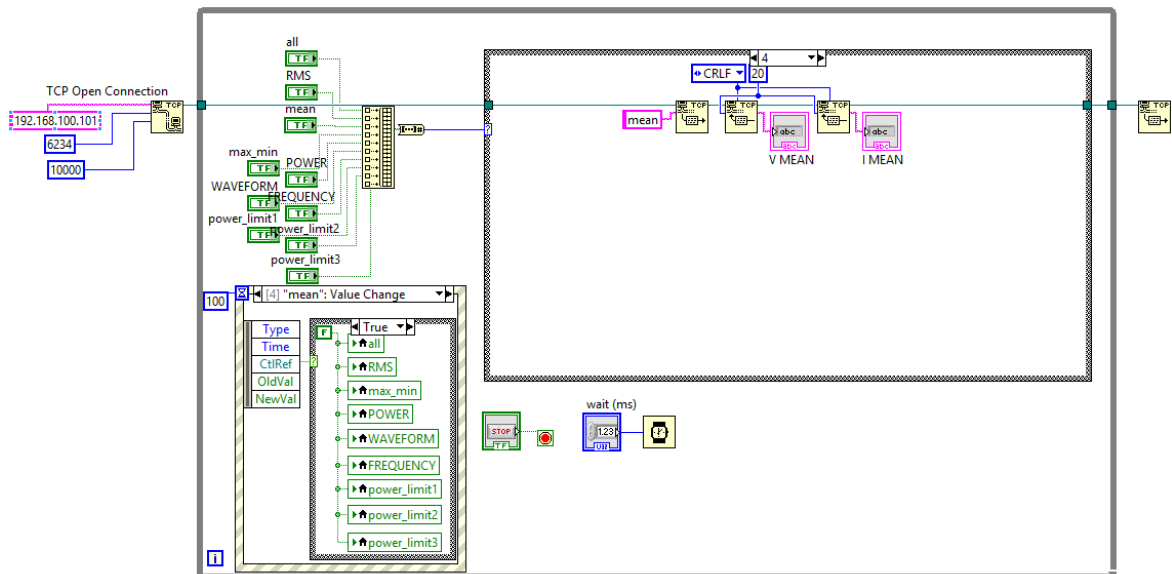
Σχήμα 4.7: Η περίπτωση WAVEFORM

Για την εμφάνιση των μέγιστων και ελαχίστων τιμών της τάσης και του ρεύματος πρέπει να επιλεγεί το κουμπί "MAX-MIN". Σε αυτή την περίπτωση εκτελείται ο κώδικας του σχήματος 4.8. Οι τιμές που εμφανίζονται όταν εκτελείται ο συγκεκριμένος κώδικας είναι τέσσερις, η μέγιστη τιμή της τάσης, η ελάχιστη τιμή της τάσης, η μέγιστη τιμή του ρεύματος και η ελάχιστη τιμή του ρεύματος.



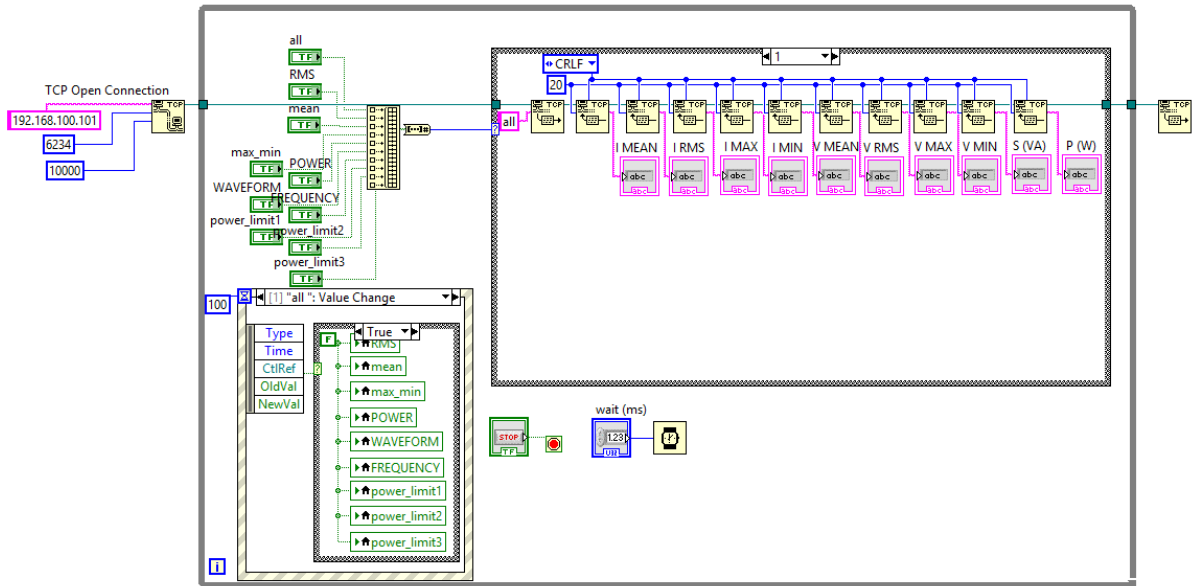
Σχήμα 4.8: Η περίπτωση MAX-MIN.

Όταν είναι επιθυμητή η εμφάνιση της μέσης τιμής της τάση και της μέσης τιμής του ρεύματος αρκεί να πατηθεί το κουμπί “MEAN”. Τότε ξεκινά η εκτέλεση του κώδικα που παρουσιάζεται στο σχήμα 4.9. Το μήνυμα που αποστέλλεται στον μικροελεγκτή είναι το “max-min” με αποτέλεσμα ο μικροελεγκτής να επιστρέφει την μέση τιμή της τάσης και την μέση τιμή του ρεύματος.



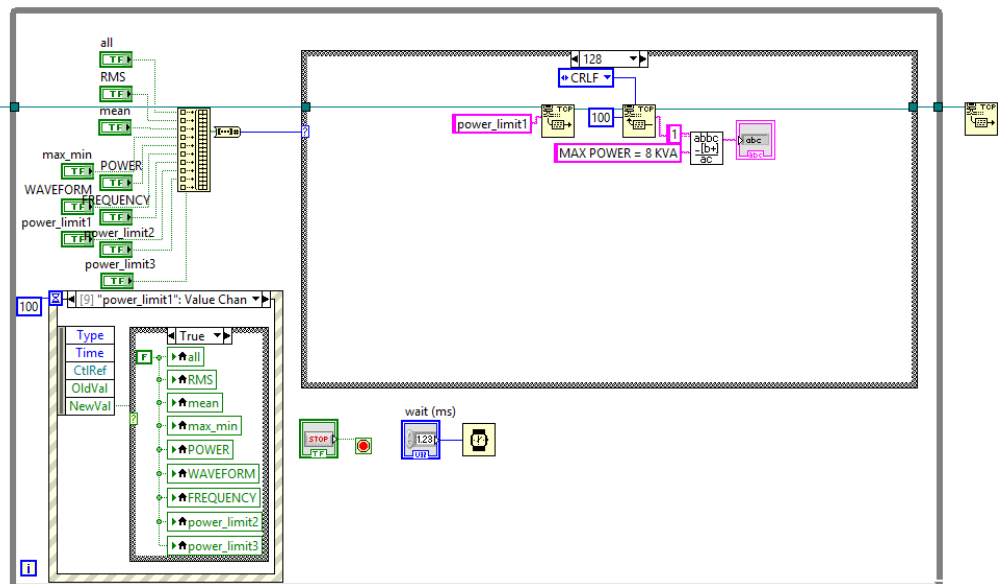
Σχήμα 4.9: Η περίπτωση MEAN.

Για την εμφάνιση των μέγιστων τιμών, ελαχίστων τιμών, των μέσων τιμών και των ενεργών τιμών της τάσης και του ρεύματος καθώς και των τιμών της φαινόμενης και ενεργούς ισχύος ταυτόχρονα, έχει αναπτυχθεί ο κώδικας του σχήματος 4.10. Ο κώδικας αυτός εκτελείται όταν πατηθεί το κουμπί «ALL». Σε αυτή την περίπτωση, το μήνυμα που αποστέλλεται στον μικροελεγκτή είναι το «all», και οι τιμές που επιστρέφει ο μικροελεγκτής είναι οι παραπάνω.

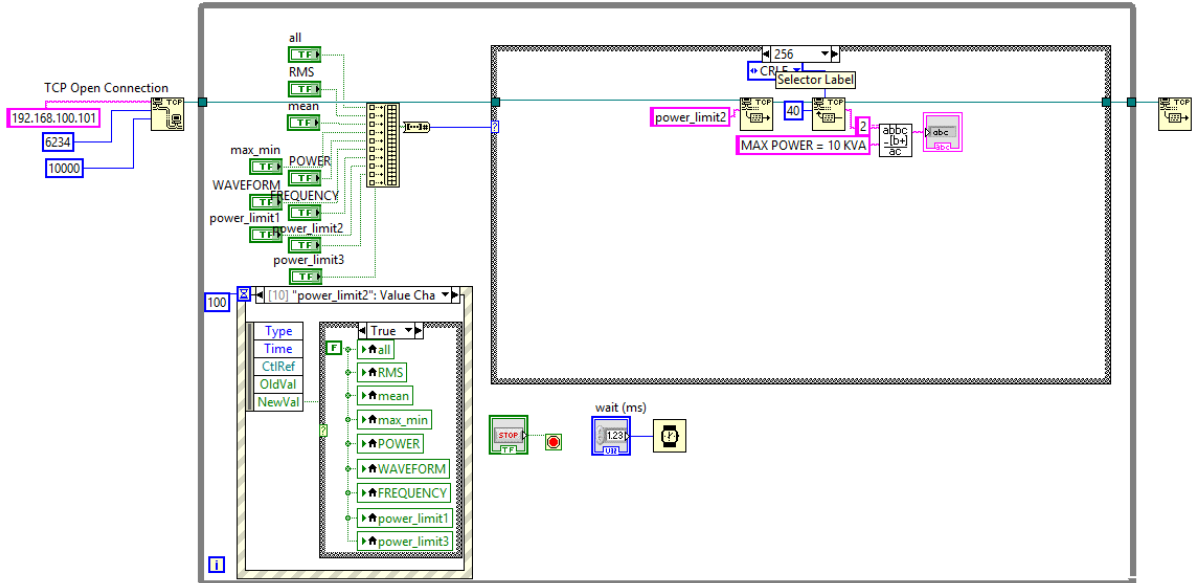


Σχήμα 4.10: Η περίπτωση ALL.

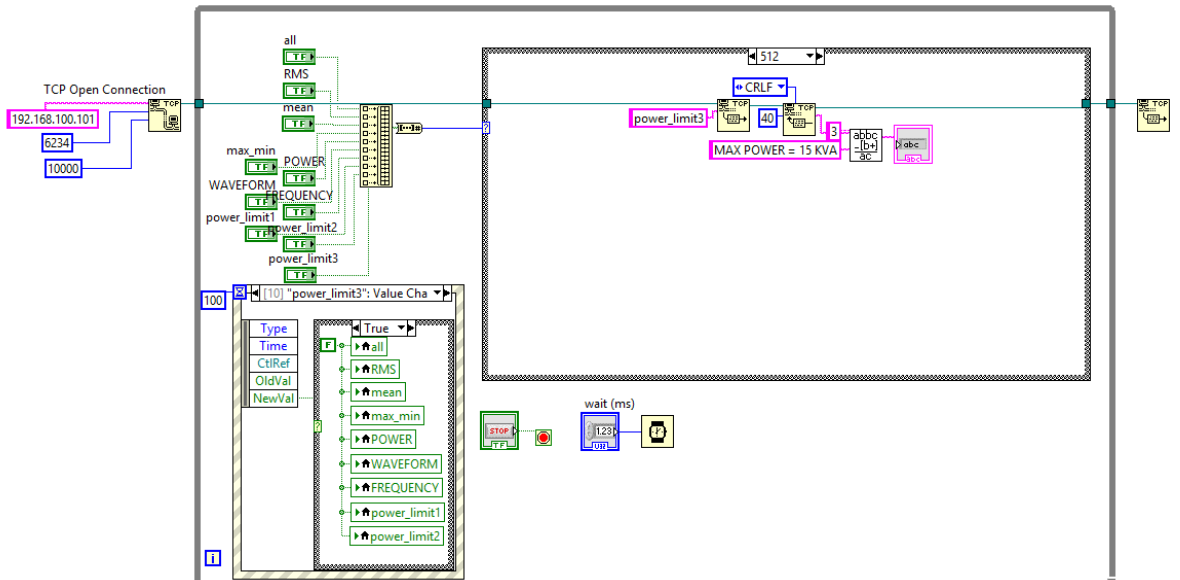
Τέλος για την επιλογή του ορίου της καταναλισκόμενης ισχύος εκτελείται ο κώδικας που παρουσιάζεται στα σχήματα 4.11, 4.12 και 4.13. Όταν πατηθεί το κουμπί “power limit 1” εκτελείται ο κώδικας του σχήματος 4.11 στέλνοντας στο μικροελεγκτή το μήνυμα “power_limit_1”. Σε αυτή την περίπτωση ο μικροελεγκτής επιστρέφει τον αριθμό 1 με σκοπό την επιβεβαίωση της λήψης του νέου ορίου. Με αντίστοιχο τρόπο πραγματοποιείται και ο καθορισμός των άλλων δυο διαθέσιμων ορίων.



Σχήμα 4.11: Η περίπτωση power limit 1



Σχήμα 4.12: Η περίπτωση power limit 2.



Σχήμα 4.13: Η περίπτωση power limit 3

Κεφάλαιο 5

Αποτελέσματα – Συμπεράσματα.

5.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα κατά την λειτουργία του συγκεκριμένου αναλυτή ηλεκτρικής ενέργειας που αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας και για τις δύο περιπτώσεις λειτουργίας (τοπική λειτουργία και απομακρυσμένη λειτουργία). Η κατάλληλη τοποθεσία σύνδεσης του συγκεκριμένου αναλυτή θα μπορούσε να είναι στον ηλεκτρολογικό πίνακα μιας ηλεκτρολογικής εγκατάστασης οπου θα γινόταν και ο χωρισμός των μη κρίσιμων φορτίων και θα ήταν δυνατός ο υπολογισμός της συνολικής καταναλισκόμενης ισχύος.

Η ολοκληρωμένη διάταξη που παρουσιάζεται στο σχήμα 5.1 εκτός από την πλατφόρμα Arduino, το Ethernet Shield, την LCD οθόνη, τον αισθητήρα ρεύματος και την διάταξη μέτρησης της τάσης περιλαμβάνει και ένα διπολικό μικροαυτόματο ο οποίος προστατεύει το κύλημα ισχύος σε περίπτωση υπερέντασης ή βραχυκυκλώματος. Επίσης περιλαμβάνει δυο ρελέ ισχύος, δύο ρευματοδότες, τέσσερα LED με διαφορετικά χρώματα τα οποία ενεργοποιούνται σε διαφορετικές καταστάσεις και έναν διακόπτη για reset. Το LED με



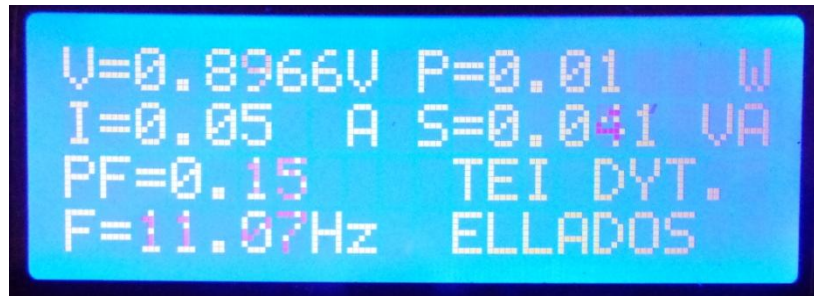
Σχήμα 5.1: Ο αναλυτής ενέργειας

πράσινο χρώμα ενεργοποιείται όταν ο μικροελεγκτής έχει αρχίσει την εκτέλεση του προγράμματος, το LED με κίτρινο χρώμα ενεργοποιείται όταν η τιμή του συντελεστή ισχύος είναι μικρότερη από το προκαθορισμένο όριο του 0,8. Ενώ το LED με κόκκινο χρώμα ενεργοποιείται όταν η καταναλισκόμενη ισχύς έχει υπερβεί το προκαθορισμένο όριο. Τέλος το LED με μπλε χρώμα ενεργοποιείται όταν ο μικροαυτόματος έχει διακόψει την τροφοδοσία του κυκλώματος ισχύος. Η διάταξη διαθέτει επίσης τρεις θύρες εισόδου. Η μια αντιστοιχεί στην γενική τροφοδοσία στην οποία πραγματοποιούνται οι μετρήσεις των ηλεκτρικών μεγεθών. Η δεύτερη είναι θύρα RJ-45 οπου πραγματοποιείται η σύνδεση του

Ethernet Shield με το router μέσω καλωδίου Ethernet. Ενώ η τρίτη θύρα είναι μια USB θύρα η οποία αξιοποιείται για την τροφοδοσία της πλατφόρμας Arduino και για τη φόρτωση του προγράμματος στον μικροελεγκτή.

5.2 Τοπική λειτουργία

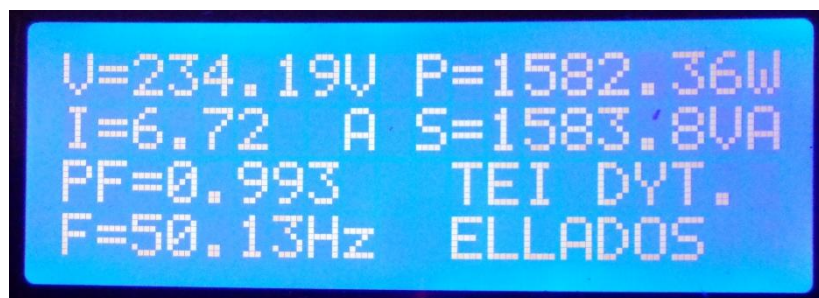
Όταν ο χρήστης δεν είναι συνδεδεμένος απομακρυσμένα τότε ο αναλυτής λειτουργεί τοπικά. Σε αυτή την περίπτωση ο μικροελεγκτής εκτελεί το μέρος του προγράμματος που αντιστοιχεί στην τοπική λειτουργία και εμφανίζει τα αποτελέσματα στην ενσωματωμένη LCD οθόνη όπως φαίνεται στο σχήμα 5.2.



Σχήμα 5.2: Η παρουσίαση των μετρήσεων στην LCD οθόνη.

Παρόλο που στην συγκεκριμένη περίπτωση δεν έχει συνδεθεί η παροχή ισχύος οι μετρούμενες τιμές δεν είναι μηδέν λόγω της παρουσίας ορισμένων σφαλμάτων που οφείλονται κυρίως στην τάση αναφοράς του ADC. Η τιμή της συχνότητας δεν έχει κανένα νόημα σε αυτή την περίπτωση, εφόσον στην διάταξη δεν έχει συνδεθεί κανένα σήμα προς μέτρηση.

Στο σχήμα 5.3 παρουσιάζονται τα αποτελέσματα που εμφανίζονται στην περίπτωση που ο αναλυτής έχει συνδεθεί στο δίκτυο της ΔΕΗ και τροφοδοτεί ένα ωμικό φορτίο. Στην συγκεκριμένη περίπτωση η μετρούμενη τάση έχει τιμή 234.19V, ενώ το ρεύμα που διαρρέει το φορτίο έχει τιμή 6,72A. Και αφού είναι ωμικό φορτίο η τιμή του συντελεστή ισχύος είναι σχεδόν μονάδα.



Σχήμα 5.3: Η παρουσίαση των μετρήσεων στην LCD οθόνη.

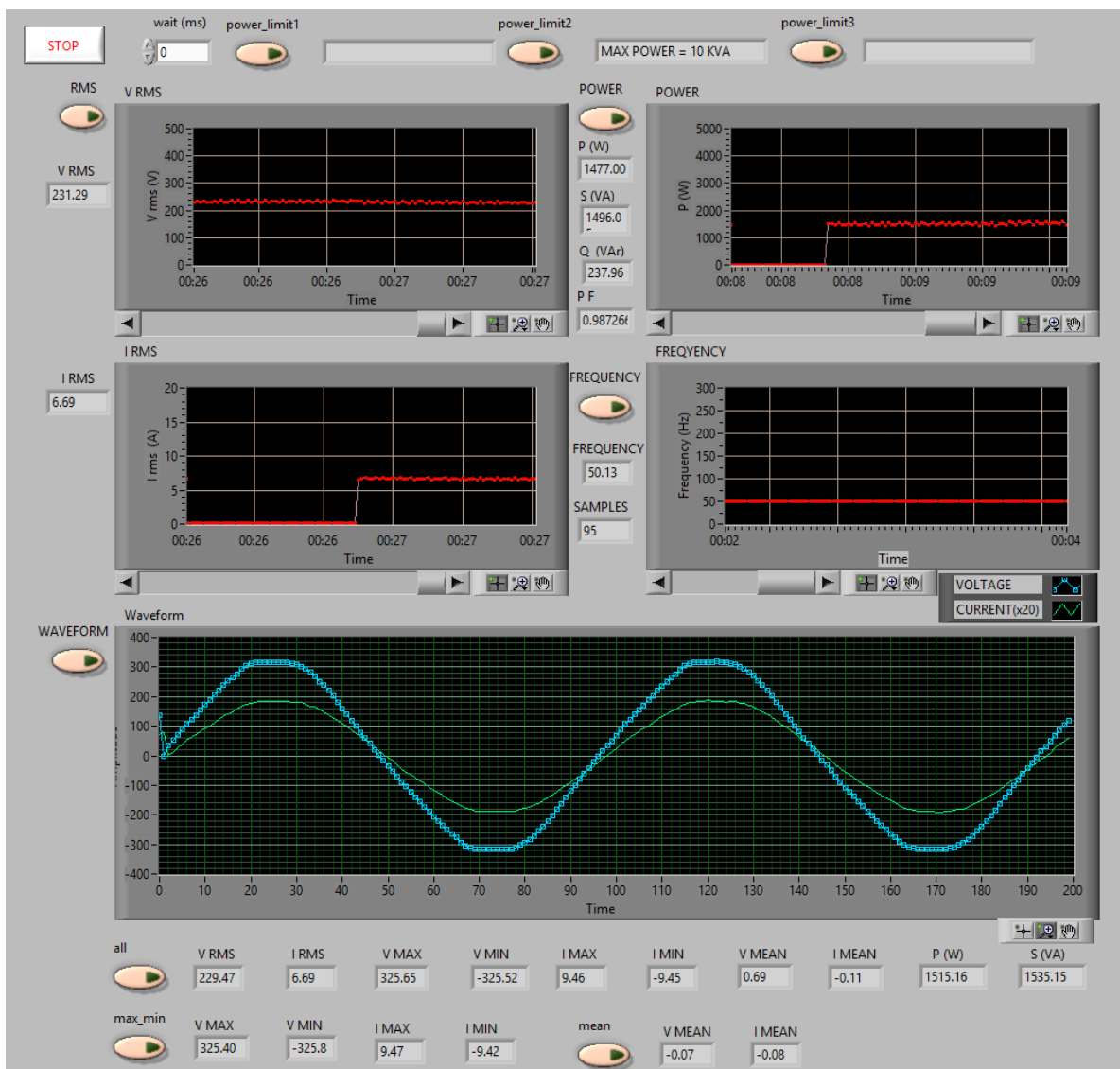
Όταν ως ηλεκτρικό φορτίο συνδεθεί ένας μονοφασικός κινητήρας φορτίο τότε τα αποτελέσματα των μετρήσεων στην συγκεκριμένη περίπτωση παρουσιάζονται στο σχήμα 5.4. Όπως φαίνεται και στο συγκεκριμένο σχήμα η τιμή του συντελεστή ισχύος είναι χαμηλή εξαιτίας του φορτίου που έχει συνδεθεί, με αποτέλεσμα η τιμή της φαινόμενης ισχύος να είναι μεγαλύτερη από την τιμή της πραγματικής ισχύος.



Σχήμα 5.4: Η παρουσίαση των μετρήσεων στην LCD οθόνη.

5.3 Απομακρυσμένη λειτουργία

Με την έναρξη της εφαρμογής που αναπτύχθηκε στο LabVIEW ενεργοποιείται το Front Panel του GUI. Με την εκτέλεση του GUI, αρχικά πραγματοποιείται η σύνδεση του GUI με τον μικροελεγκτή και στην συνέχεια ο χρήστης μπορεί να πατήσει ένα από τα διαθέσιμα

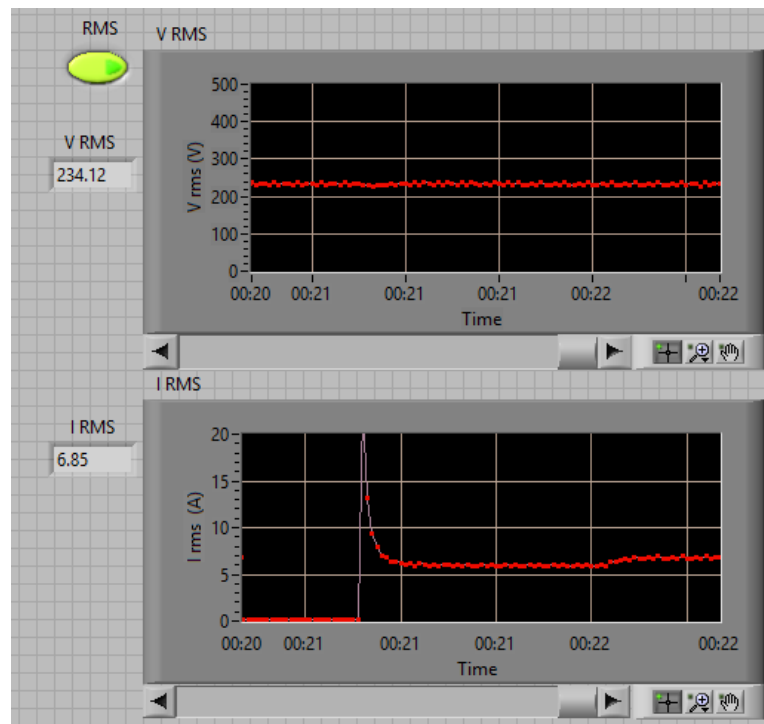


Σχήμα 5.5: Το Front Panel της εφαρμογής.

κουμπιά. Σε αυτή την περίπτωση ο μικροελεγκτής εκτελεί το μέρος του προγράμματος το οποίο αντιστοιχεί στην απομακρυσμένη λειτουργία, τυπώνοντας στην LCD οθόνη το μήνυμα "REMOTE CONTROL IS ON". Κατά την απομακρυσμένη λειτουργία της

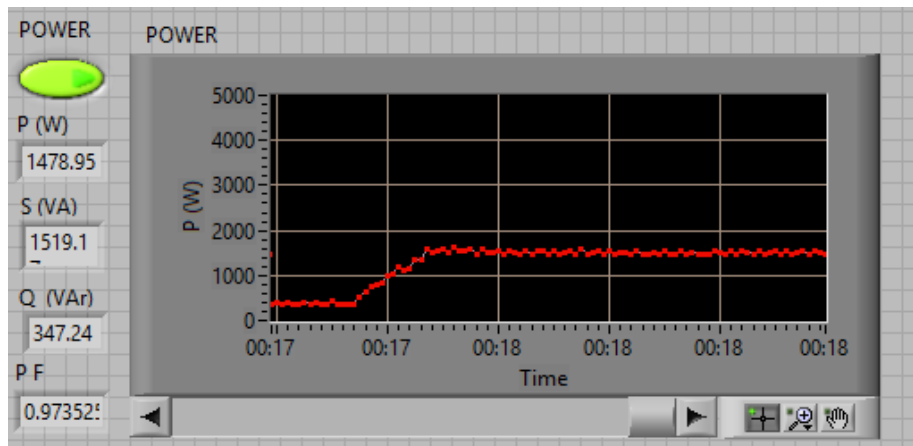
εφαρμογής ο χρήστης έχει την δυνατότητα να επιλέξει το επιθυμητό κουμπί και να παρατηρεί τα αντίστοιχα αποτελέσματα. Μια εικόνα του Front Panel όπου οι περισσότερες λειτουργίες έχουν ενεργοποιηθεί για μια φορά τουλάχιστον παρουσιάζεται στο σχήμα 5.5.

Ποιο αναλυτικά όταν γίνει ενεργοποίηση του Control με την ονομασία «RMS» τότε ο μικροελεγκτής εκτελεί το μέρος του προγράμματος το οποίο στέλνει της ενεργές τιμές της τάσης και του ρεύματος με αποτέλεσμα να πραγματοποιείται η εμφάνιση τους όπως φαίνεται στο σχήμα 5.6. Με την επιλογή αυτή ο χρήστης παρατηρεί ταυτόχρονα τις τιμές αυτές αλλά και τις μεταβολές τους μέσω των γραφικών παραστάσεων. Στην συγκεκριμένη περίπτωση η τιμή της τάσης είναι σταθερή στα 234.12 V ενώ για την τιμή του ρεύματος παρατηρείται μια απότομη αύξηση και στην συνέχεια σταθεροποιείται στα 6,85 A. Η απότομη μεταβολή της τιμής του ρεύματος στην συγκεκριμένη περίπτωση ωφελείτε στην εκκίνηση ενός μονοφασικού κινητήρα.



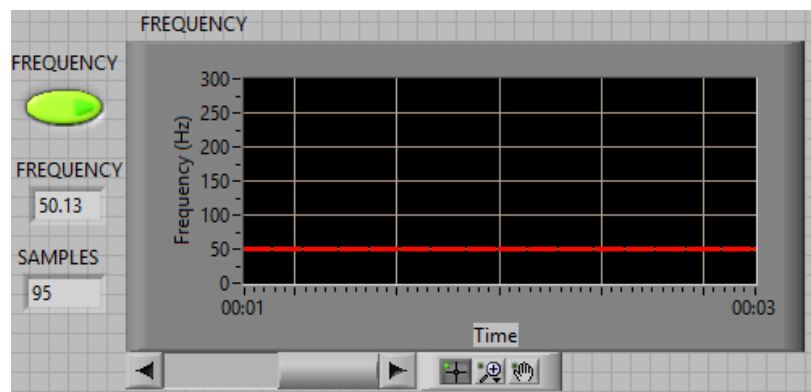
Σχήμα 5.6: Οι τιμές και η γραφική αναπαράσταση των ενεργών τιμών τάσης και ρεύματος

Ενώ όταν επιλεχθεί το κουμπί με την ονομασία «POWER» τότε αυτόματα σταματά η απεικόνιση των ενεργών τιμών και ξεκινά η αναπαράσταση των τιμών της ενεργού ισχύος, της άεργου ισχύος, της φαινόμενης ισχύος και του συντελεστή ισχύος όπως φαίνεται και στο σχήμα 5.7. Σε αυτή την περίπτωση επιλέχτηκε από τις παραπάνω τιμές η τιμή της ενεργούς ισχύος αναπαρασταθεί και σε γραφική παράσταση ώστε να παρατηρείται με ευκολία η μεταβολή της κατανάλωσης των συνδεδεμένων ηλεκτρικών φορτίων. Στο παράδειγμα του σχήματος 5.7 παρουσιάζεται η μεταβολή της τιμής της ενεργού ισχύος ενός ηλεκτρικού φορτίου.



Σχήμα 5.7: Η κυματομορφή την ενεργούς ισχύος.

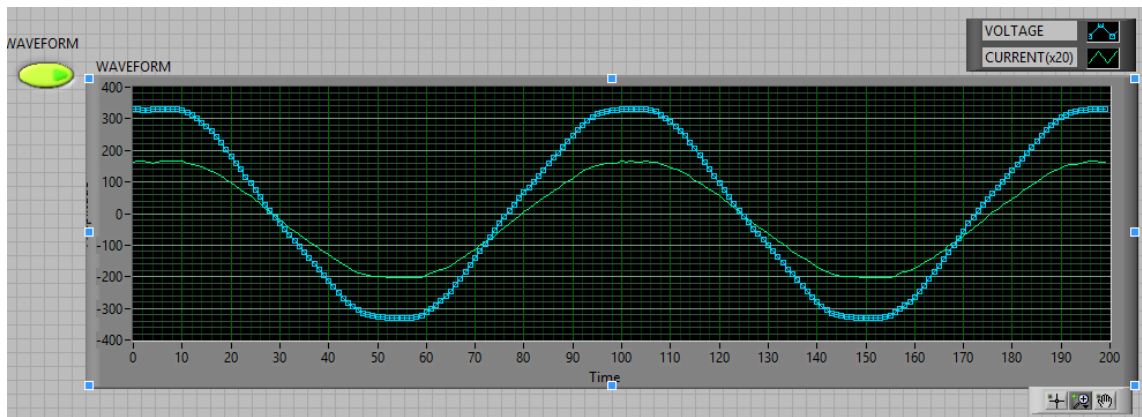
Όταν επιλεχθεί το κουμπί με την ονομασία «FREQUENCY» τότε ξεκινά η αναπαράσταση των τιμών της συχνότητας της τάσης του δικτύου και ο αριθμός των δειγμάτων που λαμβάνει ο μικροελεγκτής σε μια περίοδο. Η συχνότητα εμφανίζεται και στην γραφική παράσταση όπως φαίνεται στο σχήμα 5.6. Στο συγκεκριμένο παράδειγμα η τιμή της συχνότητας είναι σταθερή στα 50,13 Hz, αυτό οφείλεται στο γεγονός ότι ο αναλυτής έχει συνδεθεί στο δίκτυο της ΔΕΗ που έχει ονομαστική τάση 230V και συχνότητα 50 Hz.



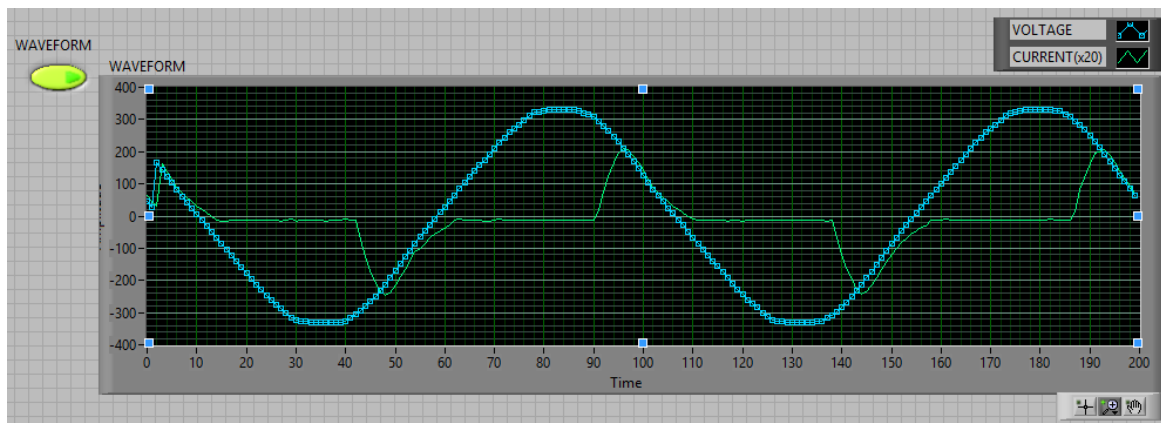
Σχήμα 5.8: Η κυματομορφή της συχνότητας

Για να γίνει η απεικόνιση των στιγμιαίων τιμών της τάσης και του ρεύματος πρέπει να επιλεχθεί το κουμπί με την ονομασία «WAVEFORM». Σε αυτή την περίπτωση ο μικροελεγκτής στέλνει τους τιμές (τάσης – ρεύματος) που έχει αποθηκεύσει σε δυο πινάκες. Η αναπαράσταση των τιμών αυτών απεικονίζονται στην γραφική παράσταση του σχήματος 5.9. Η κυματομορφή με το μπλε χρώμα αντιστοιχεί στην τάση του δικτύου, ενώ η κυματομορφή με το πράσινο χρώμα αντιστοιχεί στο ρεύμα του φορτίου.

Για την καλύτερη αναπαράσταση του ρεύματος σε σχέση με την τάση του δικτύου έχει γίνει πολλαπλασιασμός των τιμών του ρεύματος με τον συντελεστή 20. Στο παράδειγμα του σχήματος 5.9 έχει χρησιμοποιηθεί ωμικό φορτίο και για τον λόγο αυτό οι κυματομορφές της τάσης και του ρεύματος να είναι συμφασικές. Ενώ στο παράδειγμα του σχήματος 5.10 παρουσιάζονται οι κυματομορφές της τάσης και του ρεύματος ενός φορτίου που κάνει χρήση ηλεκτρονικών ισχύος.

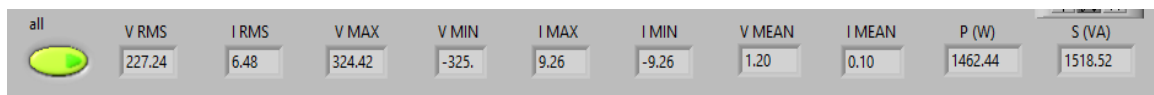


Σχήμα 5.9: Η κυματομορφή της τάσης και του ρεύματος για ωμικό φορτίο



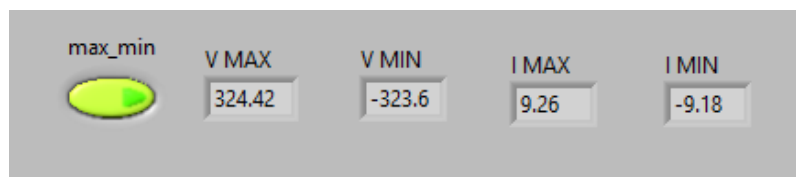
Σχήμα 5.10: Η κυματομορφή της τάσης και του ρεύματος για φορτίο που περιέχει και ηλεκτρονικά ισχύος

Για την ταυτόχρονη παρατήρηση των υπολογιζόμενων ηλεκτρικών μεγεθών αρκεί να ενεργοποιηθεί το κουμπί με την ονομασία «ALL». Η επιλογή αυτή εμφανίζει τις ενεργές τιμές, τις μέγιστες τιμές, τις ελάχιστες τιμές, τις μεσαίες τιμές της τάσης και του ρεύματος καθώς και την τιμή της ενεργούς και της φαινόμενης ισχύος όπως φαίνεται στο σχήμα 5.11.



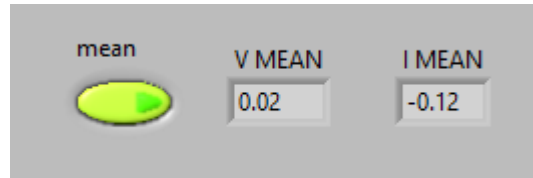
Σχήμα 5.11: Η επιλογή "ALL"

Ενώ όταν είναι επιθυμητή μόνο η παρατήρηση των μέγιστων και των ελαχίστων τιμών της τάσης και του ρεύματος τότε πρέπει να ενεργοποιηθεί το κουμπί με την ονομασία «MAX-MIN». Ενεργοποιώντας αυτό το κουμπί πραγματοποιείται η εμφάνιση μόνο των μέγιστων αι ελαχίστων τιμών της τάσης και του ρεύματος όπως φαίνεται και στο σχήμα 5.12.



Σχήμα 5.12: Η επιλογή "MAX-MIN"

Τέλος όταν επιλεγεί το κουμπί με την ονομασία «MEAN» εμφανίζει μόνο την μέση τιμή της τάσης και την μέση τιμή του ρέματος. Όπως παρουσιάζεται στο σχήμα 5.13 οι τιμές που εμφανίζονται σε αυτή την περίπτωση είναι σχεδόν μηδενικές. Αυτό οφείλεται στο γεγονός ότι ο αναλυτής ενέργειας έχει συνδεθεί στο δίκτυο της ΔΕΗ της οποίας η τάση έχει ημιτονοειδή κυματομορφή με αποτέλεσμα η μέση τιμή της να είναι 0V.



Σχήμα 5.13: Η επιλογή "MEAN"

Μια επιπλέον διαθέσιμη λειτουργία που είναι διαθέσιμη κατά την απομακρυσμένη λειτουργία είναι η επιλογή του ορίου της καταναλισκόμενης ισχύος. Οι διαθέσιμες επιλογές είναι τρεις, 8 KVA, 10KVA, 15KVA οι οποίες ενεργοποιούνται από τα Control 'power_limit1', 'power_limit2' και 'power_limit3' αντίστοιχα. Όταν καμία από αυτές τις επιλογές δεν επιλεγεί τότε το προκαθορισμένο όριο είναι 5 KVA. Όταν η τιμή της καταναλισκόμενης ισχύς είναι μικρότερη από το επιλεγμένο όριο τότε τροφοδοτούνται όλα τα συνδεδεμένα φορτία. Όταν η τιμή αυτή υπερβεί το επιλεγόμενο όριο διακόπτεται η τροφοδοσία των μη κρίσιμων φορτίων.

Στο GUI υπάρχουν δυο επιπλέον κουμπιά, το STOP και το Wait (ms). Το κουμπί STOP χρησιμοποιείται για τον τερματισμό της λειτουργίας του GUI και συνεπώς απενεργοποιεί την απομακρυσμένη λειτουργία του αναλυτή. Το Control Wait (ms) δίνει την δυνατότητα στον χρήστη να επιλέξει την ταχύτητα με την οποία θα ανανεώνονται οι τιμές στο GUI.

5.4 Συμπεράσματα – Προτάσεις

Η παρούσα πτυχιακή εργασία είχε ως αντικείμενο τον σχεδιασμό και την ανάπτυξη ενός αναλυτή ηλεκτρικής ενέργειας ο οποίος παρέχει στον χρήστη την δυνατότητα να ελέγχει τον αναλυτή ενέργειας και να παρατηρεί τα μεγέθη που υπολογίζονται είτε στο σημείο που πραγματοποιούνται οι μέτρησεις (τοπική λειτουργία) είτε απομακρυσμένα μέσω του GUI (απομακρυσμένη λειτουργία). Η ανάπτυξη του συγκεκριμένου αναλυτή βασίστηκε στην πλατφόρμα Arduino UNO η οποία διαθέτει ένα μικροελεγκτή ATmega 328. Για την ανάπτυξη του προγράμματος που φορτώθηκε στον μικροελεγκτή αξιοποιήθηκε το Arduino IDE ενώ για την ανάπτυξη του GUI για τον απομακρυσμένο έλεγχο του αναλυτή ενέργειας χρησιμοποιήθηκε το λογισμικό πακέτο LabVIEW.

Όπως παρατηρήθηκε και από τις μέτρησεις που πραγματοποιηθήκαν τα αποτελέσματα των υπολογισμών που πραγματοποιεί ο μικροελεγκτής παρουσιάζουν κάποιο σφάλμα το οποίο οφείλεται σε συγκεκριμένες αιτίες. Πιο συγκεκριμένα ένα μεγάλο ποσοστό του σφάλματος οφείλεται στην μη σταθερότητα της τάσης αναφοράς που χρησιμοποιείται από τον ADC του μικροελεγκτή ATmega328. Η τάση η οποία παρέχεται στην πλατφόρμα Arduino από έναν Η/Υ μέσω της θύρας USB, χρησιμοποιείται επίσης και ως τάση αναφοράς του ADC. Όταν η τιμή της συγκεκριμένης τάσης μεταβάλλεται τότε και τα αποτελέσματα των μετρήσεων μεταβάλλονται ενώ στην πραγματικότητα το μετρούμενο μέγεθος είναι σταθερό. Για την αντιμετώπιση του παραπάνω προβλήματος μπορεί να χρησιμοποιηθεί μια εξωτερική ηλεκτρονική διάταξη με σκοπό την σταθεροποίηση της τάσης αναφοράς σε επιθυμητά επίπεδα ακρίβειας. Επομένως ο ADC θα πρέπει να ρυθμιστεί κατάλληλα ώστε να αξιοποιεί την εξωτερική τάση αναφοράς και όχι την τάση τροφοδοσίας του Arduino.

Επίσης ένα πρόσθετο σφάλμα οφείλεται στην διάταξη για την προσαρμογή της τάσης του δικτύου ώστε να είναι δυνατή δειγματοληψία της τάσης δικτύου από τον ADC του ATmega328. Ο συγκεκριμένος ADC δεν έχει την δυνατότητα μέτρησης αρνητικών τιμών με αποτέλεσμα να είναι απαραίτητη η αξιοποίηση μιας DC συνιστώσας ώστε να εξαιρεθούν οι αρνητικές τιμές της τάσης. Στην συγκεκριμένη υλοποίηση η DC συνιστώσα προέρχεται από την τάση τροφοδοσίας του Arduino που όπως αναφέρθηκε και παραπάνω δεν είναι απόλυτα σταθερή. Για να αντιμετωπιστεί το πρόβλημα αυτό θα μπορούσε και σε αυτή την περίπτωση να αξιοποιηθεί μια εξωτερική τάση που θα προέρχεται από κάποιο κύκλωμα σταθεροποίησης. Τέλος για την προσαρμογή της τάσης του δικτύου χρησιμοποιήθηκε και ένας μετασχηματιστής υποβιβασμού. Το γεγονός αυτό έχει ως αποτέλεσμα την αλλοίωση του αρχικού σήματος λόγω του φαινομένου της μαγνητικής υστέρησης και των ανώτερων αρμονικών που προστίθενται στο αρχικό σήμα.

Ο συγκεκριμένος αναλυτής θα μπορούσε να επεκταθεί μελλοντικά με την αξιοποίηση ενός εξωτερικού ADC με μεγαλύτερη ταχύτητα δειγματοληψίας και ακρίβεια ώστε να αξιοποιηθεί για την ανάλυση των ανώτερων αρμονικών που παρουσιάζονται σε μια ηλεκτρική εγκατάσταση.

Μελλοντικά θα μπορούσε να αναπτυχθεί ένας κεντρικός σταθμός συλλογής δεδομένων που θα συλλέγει μετρήσεις από ένα σύνολο αναλυτών ενέργειας με δυνατότητες αντίστοιχες με του αναλυτή ενέργειας που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας. Τα δεδομένα που θα συλλέγονται θα μπορούσαν να αξιοποιηθούν ώστε να υπολογίζεται η χρέωση του καταναλωτή κάθε χρονική στιγμή. Έτσι ο καταναλωτής είναι σε θέση να ενημερώνεται διαρκώς για την κατανάλωση της ηλεκτρικής ενέργειας και να προσαρμόζει αντίστοιχα τις εκάστοτε ανάγκες του. Επίσης αφού οι πάροχοι ηλεκτρικής ενέργειας θα γνωρίζουν την συνολική κατανάλωση ισχύος θα μπορούσε η τιμολόγηση της ηλεκτρικής ισχύος να πραγματοποιείται με βάση την ζήτηση. Το γεγονός αυτό θα είχε ως αποτέλεσμα την αισθητή μείωση της κατανάλωσης κατά τις ώρες αιχμής.

Τέλος θα μπορούσε να τροποποιηθεί κατάλληλα η ηλεκτρολογική εγκατάσταση ενός σπιτιού ή οικήματος ώστε να υποστηρίζονται ξεχωριστές γραμμές τροφοδοσίας για τα μη κρίσιμα ηλεκτρικά φορτία. Σε αυτή την περίπτωση και με κατάλληλη επέκταση του συγκεκριμένου αναλυτή θα μπορούσε ο χρήστης να προκαθορίζει μέσω του αναλυτή ενέργειας τις ώρες της ημέρας που θα επιτρέπεται η τροφοδότηση των επιλεγμένων φορτίων. Συνεπώς θα μπορούσε να επιλεγεί η τροφοδότηση των παραπάνω φορτίων κατά τις χρονικές περιόδους που η χρέωση της ηλεκτρικής ενέργειας είναι μειωμένη.

Βιβλιογραφία

- [1] ATMEL, “Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash”, 8271G–AVR. February 2013.
- [2] Playground Liquid Crystal, <http://arduino.cc/en/Reference/LiquidCrystal>, τελευταία επίσκεψη, 21/10/2015.
- [3] Arduino playground Ethernet, <http://arduino.cc/en/Reference/Ethernet>, τελευταία επίσκεψη, 14/9/2015.
- [4] Arduino playground SPI, <http://arduino.cc/en/Reference/SPI>, τελευταία επίσκεψη, 14/9/2015.
- [5] Arduino Ethernet Shield, <https://www.arduino.cc/en/Main/ArduinoEthernetShield>, τελευταία επίσκεψη, 25/11/2015.
- [6] Arduino UNO, <https://www.arduino.cc/en/Main/ArduinoBoardUno>, τελευταία επίσκεψη, 25/11/2015.
- [7] H. Timmis, “*Practical Arduino Enginnerin*”, Apress, 2011.
- [8] M. Banzi, “*Getting Started with Arduino*”, Second Edition, O’REILY 2011.
- [9] J. Floyd Kelly and H. Timis, “*Arduino Adventures*”, Apress, 2013.
- [10] E. Premeaux and B. Evans, “*Arduino Projects to Save the World*”, Apress, 2011.
- [11] J. D. Warren, Josh Adams and H. Molle “*Arduino Robotics*”, Apress, 2011.
- [12] T. Karvinen and K. Karvinen “*Make: Arduino Bots and Gadgets*”, Maker Media Inc, 2011.
- [13] Newbiehack.com, Reading Multiple ADC Channels on the AVR Atmega32, <https://newbiehack.com/MicrocontrollersADCReadingMultipleAnalogVolageSources.aspx>, τελευταία επίσκεψη, 25/11/2015.
- [14] Open Energy Monitor, Measuring AC Voltage with an AC to AC power adapter, <http://openenergymonitor.org/emon/buildingblocks/measuring-voltage-with-an-acac-power-adapter>, τελευταία επίσκεψη, 25/11/2015.
- [15] Gammon Forum, Interrupts, <http://gammon.com.au/interrupts>, τελευταία επίσκεψη, 25/11/2015.
- [16] Instructables, Arduino Frequency Detection, <http://www.instructables.com/id/Arduino-Frequency-Detection/step3/Sine-Wave-Frequency-Detection/> τελευταία επίσκεψη, 25/11/2015.
- [17] Γ. Κορέτσης, “*Σχεδιασμός και κατασκευή τρίκυκλου ποδηλάτου με ηλεκτρική υποβοήθηση*”, Πτυχιακή Εργασία, Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Οχημάτων, 2013.
- [18] LabVIEW, Wikipedia, <http://en.wikipedia.org/wiki/LabVIEW>, τελευταία επίσκεψη 30/8/2015.
- [19] Α. Καλαντζόπουλος και Ε. Ζυγούρης, “*Εισαγωγή στο LabVIEW*”, Διάλεξη στα πλαίσια του μαθήματος Συστήματα Επεξεργασίας Σημάτων με DSPs του ΔΠΜΣ Ηλεκτρονική και Επεξεργασία της Πληροφορίας, Πανεπιστήμιο Πατρών, 2010.
- [20] National Instruments, “*LabVIEW Fundamentals*”, 374029A-01, August 2005.

- [21] National Instruments, “Getting Started with LabVIEW”, 373427A-01, August 2005.
- [22] National Instruments, “LabVIEW User Manual”, 320999E-01, April 2003.
- [23] Κ. Ντοντος, “Ανάπτυξη Γραφικού Περιβάλλοντος Βασισμένο στο LabVIEW για την Προσομοίωση Ελεγχόμενων Ανορθωτικών Γεφυρών, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Ηλεκτρολόγων Μηχανικών Τ.Ε., 2014.
- [24] Allegro MicroSystems, “*Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*”, ACS712-DS, Rev. 15, 2013.