

2014

Σχεδίαση και υλοποίηση γεννήτριας συναρτήσεων





ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
ΤΕΙ ΗΠΕΙΡΟΥ

ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

με Θέμα:

ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΓΕΝΝΗΤΡΙΑΣ ΣΥΝΑΡΤΗΣΕΩΝ

ΤΖΙΜΑΣ ΣΠΥΡΙΔΩΝ Α.Μ. 9969

-Άρτα 2014-

ΔΗΛΩΣΗ ΠΕΡΙ ΛΟΓΟΚΛΟΠΗΣ

Όλες οι πηγές οι οποίες συντέλεσαν στο να δημιουργηθεί αυτή η εργασία και οι οποίες ανήκουν σε άλλους παρουσιάζονται στη βιβλιογραφία στο τέλος της εργασίας.

Τα υπόλοιπα γραφόμενα είναι επινόηση των γραφόντων οι οποίοι φέρουν και την καθολική ευθύνη γι' αυτό το κείμενο και δηλώνω υπεύθυνα ότι δεν υπάρχει λογοκλοπή γι' αυτό το κείμενο.

ΠΡΟΛΟΓΟΣ

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η σχεδίαση και κατασκευή μιας γεννήτριας συναρτήσεων, με την βοήθεια του μικροελεγκτή Arduino. Ο χρήστης αλληλεπιδρά με την γεννήτρια μέσω της σειριακής οθόνης του μικροελεγκτή στην οποία εμφανίζεται το μήνυμα όπου ζητείτε το είδος της κυματομορφής που πρόκειται να επιλέξει αλλά και την συχνότητα την οποία θα σχεδιαστή στον παλμογράφο που έχουμε σύνδεση με την κατασκευή.

Υπεύθυνος κατά την εκπόνηση της πτυχιακής εργασίας ήταν ο κ. Γεώργιος Τάτσης.

ΕΥΧΑΡΙΣΤΙΕΣ

Σ' αυτό το σημείο επιθυμώ να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα της πτυχιακής κ. Τάτση Γεώργιο, για την εμπιστοσύνη που μου έδειξε αναθέτοντας το συγκεκριμένο θέμα και για την υποστήριξή του όποτε αυτή του ζητήθηκε.

Επίσης, θέλω να ευχαριστήσω θερμά την οικογένεια μου και ιδιαίτερα τις αδελφές μου, τους φίλους μου και την Μαρία Καλαμπούκη που με στήριξαν όλο το χρονικό διάστημα τόσο της δημιουργίας της πτυχιακής εργασίας όσο και ολης της φοιτητικής μου ζωής.

Η παρούσα πτυχιακή εργασία είναι αφιερωμένη στην μνήμη της αγαπημένης μου γιαγιάς, Ροζαλίας Καψοκαβάδη - Χειμαριού, του αγαπημένου μου πάππου Σπυρίδων Χειμαριού και του ευεργέτη του χωρίου μου Χρηστού Πουχά όπου με τον δικό τους τρόπο ξεχωριστά με στήριξαν και πορεύομαι στην ζωή με ένα κοινό λεγόμενο των τριών:

...η γνώση και η παιδεία θα «αμείβει» πλουσιοπάροχα και η άγνοια θα «τιμωρεί» ανελέητα.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	9
---------------	---

ΚΕΦΑΛΑΙΟ 1

ΜΙΚΡΟΕΛΕΓΚΤΕΣ

1.1 Ιστορία Μικροελεγκτών.....	10
1.2 Εισαγωγή στους μικροελεγκτές.....	10
1.3 Λειτουργία μικροελεγκτή.....	12
1.4 Χαρακτηριστικά των κριτηρίων επιλογής του μικροελεγκτή.....	12
1.5 Διαδεδομένες κατηγορίες μικροελεγκτών	13
1.6 Προγραμματισμός και εργαλεία ανάπτυξης	14
1.7 Κατασκευαστές μικροελεγκτών	15

ΚΕΦΑΛΑΙΟ 2

ARDUINO

2.1 Εισαγωγή στην υπολογιστική πλατφόρμα ARDUINO	16
2.2 Εκδόσεις Arduino	18
2.3 Το λογισμικό του Arduino.....	18
2.4 Arduino Uno.....	19
2.4.1 Χαρακτηριστικά	20
2.4.2 Μέρη ενός Arduino Uno.....	21
2.4.2.1 Ακροδέκτες	22
2.4.2.2 Ενέργεια – Τροφοδοσία	22
2.4.2.3 Μνήμη	23
2.4.2.4 Προγραμματισμός.....	23
2.4.3 Περιβάλλον ανάπτυξης Arduino IDE.....	24
2.4.4 Σειριακή οθόνη.....	26
2.4.5 Εγκατάσταση προγράμματος του μικροελεγκτή Arduino.....	27

ΚΕΦΑΛΑΙΟ 3

ΥΛΟΠΟΙΗΣΗ ΚΑΤΑΣΚΕΥΗΣ

3.1 Γενική περιγραφή γεννήτριας συχνοτήτων και παλμογράφου	30
3.1.1 Γεννήτρια συχνοτήτων	30
3.1.2 Παλμογράφος	30
3.2 Συστατικά μέρη κατασκευής.....	31
3.3 Κόστος κατασκευής.....	35
3.4 Μελέτη και σχεδίαση σημάτων.....	35
3.4.1 Ημίτονο.....	37
3.4.2 Τετραγωνικό σήμα.....	40
3.4.3 Τριγωνικό σήμα.....	42
3.4.4 Sawtooth wave (Πριονωτή κυματομορφή).....	44
3.4.5 Κυματομορφή Ράμπας	46
3.5 Τεχνική Pulse Width Modulation (Διαμόρφωση Εύρους Παλμών).....	48
3.6 Υλοποίηση κατασκευής με κύκλωμα RC.....	50

ΚΕΦΑΛΑΙΟ 4

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

4.1 Λειτουργία γεννήτριας συχνοτήτων	54
4.2 Κώδικας προγράμματος και επεξήγηση.....	54

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....

57

ΒΙΒΛΙΟΓΡΑΦΙΑ.....

62

ΕΙΣΑΓΩΓΗ

Η σχεδίαση και η υλοποίηση της γεννήτριας συναρτήσεων αναπτύχθηκε σε τρεις φάσεις.

Στην πρώτη φάση αναλύθηκαν τα θέματα που αφορούν την κατανόηση και την επιλογή του υλικού μέρους της κατασκευής (μικροελεγκτής ,εξαρτήματα υλοποίησης κυκλώματος).

Ακολούθησε η δημιουργία και η σχεδίαση των κυματομορφών ,με την βοήθεια μαθηματικών τύπων και του τριγωνομετρικού κύκλου, που θα απεικονίσουμε στον παλμογράφο.

Όσον αφορά τη τρίτη φάση, τέθηκε το θέμα της υλοποίησης και επεξηγήσεις του κώδικα για τον προγραμματισμό του μικροελεγκτή.

Διάρθρωση του κειμένου.

Στο πρώτο κεφάλαιο γίνεται αναφορά στους μικροελεγκτές. Αναλύεται η ιστορία, το πώς λειτουργεί, τα χαρακτηριστικά, τα κριτήρια επιλογής και οι κατηγορίες ενός τυπικού μικροελεγκτή. Επίσης γίνεται αναφορά στα προγραμματιστικά εργαλεία του αλλά και στους πιο γνωστούς κατασκευαστές μικροελεγκτών.

Στο δεύτερο κεφάλαιο αναλύουμε τον μικροελεγκτή Arduino και πιο συγκεκριμένα την πλατφόρμα Arduino Uno,όπου χρησιμοποιήσαμε για να υλοποιήσουμε τη γεννήτρια συναρτήσεων. Στη συνέχεια αναφέρουμε τα μέρη από τα οποία αποτελείται η πλατφόρμα, αλλά και το περιβάλλον ανάπτυξης καθώς και η εγκατάστασή του.

Στο τρίτο κεφάλαιο περιγράφουμε αναλυτικά τα μέρη της κατασκευής αλλά και τη μελέτη σχεδίασης των κυματομορφών.

Στο τέταρτο κεφάλαιο επεξηγείται το προγραμματιστικό μέρος του μικροελεγκτή.

Τέλος στο πέμπτο κεφάλαιο γίνεται μια μικρή αναφορά στα αποτελέσματα της εργασίας.

1. ΜΙΚΡΟΕΛΕΓΚΤΕΣ

1.1 Ιστορία Μικροελεγκτών.

Οι μικροελεγκτές αποτελούν μέρος ενός εντυπωσιακού αριθμού προϊόντων τα οποία βρίσκονται γύρω μας. Το αυτοκίνητό μας, τα τηλεχειριστήριά μας, η τηλεόρασή μας, οι ψηφιακές κάμερες, τα κινητά τηλέφωνα, τα πλυντήριά μας είναι μερικά από αυτά. Στην ουσία δεν θα ήταν υπερβολή να πούμε ότι η χρήση μικροελεγκτών στις μέρες μας είναι καθολική και γενικά κάθε προϊόν το οποίο αλληλεπιδρά με ένα χρήστη περιλαμβάνει ένα μικροελεγκτή, ο οποίος παίζει το ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων. Δεν είναι τυχαίο πλέον, ότι πολλές βιομηχανίες προσανατολίζονται σε εφαρμογές όπου αποτελούνται από ηλεκτρικά κυκλώματα τα οποία ελέγχονται από μικροελεγκτή σε αντίθεση με την πρακτική του παρελθόντος όπου χρησιμοποιούσαν ογκώδεις ηλεκτρονικούς υπολογιστές (PC) ή ηλεκτρονόμους και σύνθετη αλλά μόνιμη λογική. Η επιθυμία των κατασκευαστών συστημάτων να δημιουργήσουν συστήματα με περισσότερες δυνατότητες και με μικρότερο μέγεθος οδήγησε στην ανάγκη για ενσωμάτωση όλων των λειτουργιών ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα. Η χρήση μικροελεγκτών ξεκίνησε πριν από περίπου τριάντα πέντε χρόνια. Το πρώτο μοντέρνο ενσωματωμένο σύστημα (embedded system) ήταν ο υπολογιστής του διαστημοπλοίου Apollon, ο οποίος αναπτύχθηκε από τον Charles Stark Draper στο Εργαστήριο Instrumentation Laboratory του MIT. Τα ηλεκτρονικά κυκλώματα πλέον μπορούσαν να απλοποιηθούν εύκολα με τη χρήση ενός μικροελεγκτή προγραμματισμένος με ένα λογισμικό ελέγχου για την αντικατάσταση αυτών των ηλεκτρονικών κυκλωμάτων που είχαν χρησιμοποιηθεί προηγουμένως για τον έλεγχο συσκευών. Η σύγχρονη τεχνολογία βασίζεται ολοένα και περισσότερο σε συστήματα μικροελεγκτών. Ο κλάδος αυτός είναι ένας από τους ταχύτερα αναπτυσσόμενους κλάδους της ηλεκτρονικής. Στις μέρες μας, οι μικροελεγκτές εμφανίζονται σχεδόν σε κάθε ηλεκτρονική συσκευή.

1.2 Εισαγωγή στους μικροελεγκτές.

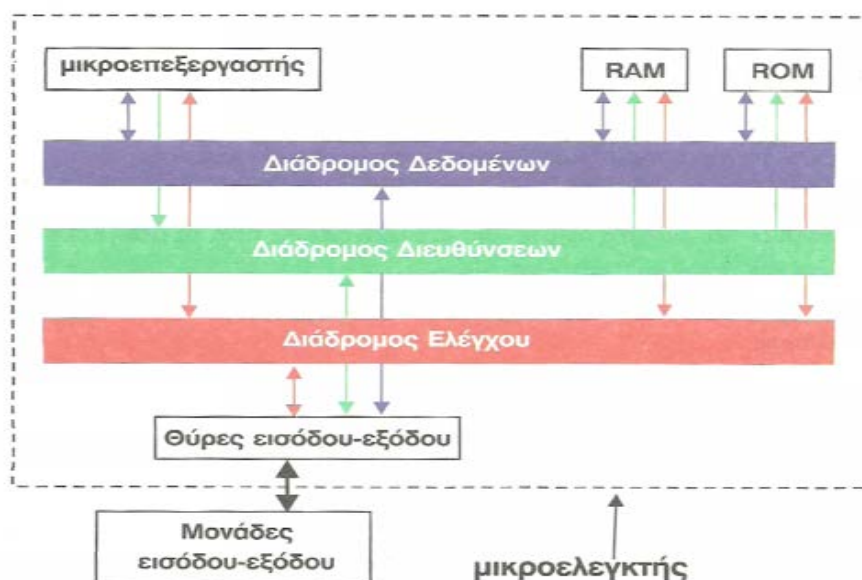
Ένας μικροελεγκτής είναι ένας ολόκληρος μικροϋπολογιστής σχεδιασμένος πάνω σε ένα και μόνο ολοκληρωμένο κύκλωμα υψηλής κλίμακας ολοκλήρωσης. Το ολοκληρωμένο αυτό κύκλωμα εμπεριέχει όχι μόνο τον επεξεργαστή αλλά και μνήμη RAM ROM, χρονιστές, πόρτες και άλλες κοινές περιφερειακές λειτουργικές μονάδες εισόδου εξόδου. Ο μικροελεγκτής όπως και ο μικροεπεξεργαστής, έχει σχεδιαστεί για να ανακαλεί δεδομένα, να κάνει πάνω σ' αυτά περιορισμένες πράξεις και με βάση αυτές τις πράξεις να ελέγχει το περιβάλλον του. Όπως κάθε υπολογιστικό κύκλωμα, περιέχει μονάδες εισόδου/εξόδου, μία κεντρική μονάδα επεξεργασίας, έναν αριθμό καταχωρητών και κυκλώματα μνήμης, εσωτερικούς χρονιστές – απαριθμητές, αριθμητική και λογική μονάδα, μνήμη προγράμματος (ROM ή EPROM). Κάθε μικροελεγκτής είναι ικανός να ανταλλάξει σήματα με το εξωτερικό περιβάλλον. Δέχεται σήματα εισόδου από διάφορους αισθητήρες, (θερμότητας πίεσης φωτός κ.τ.λ.). Εκτελεί πράξεις ανάμεσα σε μεταβλητές, καταχωρεί τιμές στη μνήμη RAM που διαθέτει και παράγει σήματα εξόδου που ελέγχουν άλλες συσκευές. Έτσι μπορεί να οδηγήσει ηλεκτρονόμους, δίοδους LED, κινητήρες και άλλα κατάλληλα κυκλώματα, που συνήθως περιλαμβάνονται σε κάθε μορφή αυτοματισμού. Έτσι με λίγα λόγια θα μπορέσαμε να πούμε

ότι ένας μικροελεγκτής είναι ένα μικρό υπολογιστικό κύκλωμα σχεδιασμένο σε ένα ολοκληρωμένο κύκλωμα υψηλής κλίμακας ολοκλήρωσης. Έχει δυνατότητα να επικοινωνεί με το εξωτερικό περιβάλλον, να στέλνει σήματα διακοπών, να εκτελεί πράξεις ανάμεσα σε μεταβλητές χρησιμοποιώντας καταχωρητές ειδικού σκοπού.

Κάθε μικροελεγκτής περιέχει τα παρακάτω στοιχεία:

- Έναν αριθμό από καταχωρητές ειδικού σκοπού όπως: καταχωρητή εργασίας, συσσωρευτή, καταχωρητή κατάστασης, μετρητή προγράμματος, καταχωρητή εντολών, καταχωρητή δείκτη.
- εσωτερικούς χρονιστές – απαριθμητές.
- αριθμητική και λογική μονάδα εντολών.
- μονάδα αποκωδικοποιήσεις εντολών.
- μνήμη προγράμματος ROM ή EPROM
- μνήμη καταχωρητών – μεταβλητών RAM
- κυκλώματα χρονισμού και ελέγχου
- παράλληλες θύρες εισόδου – εξόδου
- άλλα περιφερειακά κυκλώματα όπως: QUART

Η σχεδίαση ενός μικροϋπολογιστικού συστήματος, ενός συστήματος δηλαδή βασισμένο σε μικροεπεξεργαστή, ο οποίος ελέγχει μια πληθώρα από περιφερειακές συσκευές, περιλαμβάνει, το ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή και τις περιφερειακές συσκευές όπως μνήμες, συσκευές εισόδου/εξόδου, αισθητήρες κ.τ.λ. όπως αναφέραμε. Η σύνδεση και ο συγχρονισμός των διαφόρων συσκευών με τον μικροεπεξεργαστή καθιστά πολύπλοκη τη σχεδίαση του συστήματος. Μια λύση στο πρόβλημα θα ήταν να υπήρχε ένα ολοκληρωμένο κύκλωμα, το οποίο να ενσωματώνει κάποιες απαραίτητες περιφερειακές συσκευές. Ένα τέτοιο κύκλωμα είναι ο μικροελεγκτής (microcontroller). Παρακάτω απεικονίζεται η σχέση μεταξύ μικροεπεξεργαστή και μικροελεγκτή.



Εικόνα 1.1: Σχέση μικροεπεξεργαστή με μικροελεγκτή

1.3 Λειτουργία μικροελεγκτή.

Στην αγορά σήμερα βρίσκουμε πληθώρα τύπων μικροελεγκτών αλλά και προγράμματα που διατίθενται για μικροελεγκτες με μια γρήγορη μάλιστα διαπιστώνουμε ότι έχουν αρκετά κοινά χαρακτηριστικά στο θέμα διαχείρισης αυτών των διάφορων τύπων. Το αποτέλεσμα είναι ότι αν μάθουμε να χειριζόμαστε έναν τύπο μικροελεγκτή δεν απαιτεί επιπλέον γνώσεις ώστε να χειριστούμε και τους άλλους τύπους επίσης. Ένας μικροελεγκτής είναι μια σχετικά ταχύτατη συσκευή έτσι ώστε μια εντολή του που εκτελείτε σε αυτόν γίνεται με μεγάλη ταχύτητα. Παρακάτω αναλύετε η λειτουργική δομή ενός μικροελεγκτή ώστε να καταλάβουμε ακόμα περισσότερα πράγματα σχετικά με το θέμα των μικροελεγκτών αλλά και την συμβολή του στην υλοποίηση της γεννήτριας συχνοτήτων όπως θα δούμε στα επόμενα κεφαλαία.

Σε πρώτη φάση λοιπόν ενεργοποιείτε η τροφοδοσία με την σειρά του το μητρώο λογικού έλεγχου ενεργοποιεί τον ταλαντωτή χαλαζία. Στα πρώτα χιλιοστά του δευτερολέπτου γίνεται η φόρτιση των πυκνωτών. Όταν η στάθμη της τάσης φτάνει στη μέγιστη τιμή και η συχνότητα του ταλαντωτή χαλαζία γίνεται σταθερή, η διαδικασία της γραφής bits σε ειδικά λειτουργικά μητρώα (SFrs) ξεκινάει. Τα πάντα γίνονται σύμφωνα με το ρολόι του ταλαντωτή και έτσι τα ηλεκτρονικά αρχίζουν να δουλεύουν. Το pc ή ο μετρητής προγράμματος μηδενίζει τη διεύθυνση της μνήμης προγράμματος. Στη συνέχεια η διεύθυνση αποστέλλει τις οδηγίες στον αποκωδικοποιητή ο οποίος αποκωδικοποιεί τις εντολές και έτσι τις εκτελεί. Μετά την εκτέλεση μιας εντολής η διεύθυνση του απαρτιμμητή προγράμματος αυξάνεται κατά 1 και ως εκ τούτου στέλνει τη διεύθυνση τις επόμενης εντολής στον αποκωδικοποιητή εντολών και εκτελεί τις επόμενες οδηγίες.

1.4 Χαρακτηριστικά των κριτηρίων επιλογής του μικροελεγκτή.

Για αρχή γνωρίσαμε σε γενικά πλαίσια τόσο τις δομές όσο και την λειτουργία του μικροελεγκτή το επόμενο βήμα είναι να αναφέρουμε τι είναι αυτό που των κάνει να είναι ξεχωριστό και μια συσκευή ως επιλογή για την χρήση του έναντι της χρήσης των επιμέρους στοιχείων που το απαρτίζουν. Τα βασικά χαρακτηριστικά και τα θετικά κριτήρια είναι τα εξής:

- **Χαμηλό κόστος.** Είναι από τα βασικότερα κριτήρια αγοράς όπου οφείλετε στον ανταγωνισμό μεταξύ των εταιριών ηλεκτρονικών αφού αυξήθηκε η ποιότητα και μειώθηκε η τιμή του.
- **Μικρό μέγεθος.** Η ολοκλήρωση των βασικών στοιχείων από τα όποια απαρτίζεται μείωσε τις διαστάσεις σε σχέση με τη χρήση των επιμέρους στοιχείων ως σύνολο.
- **Χαμηλή ισχύεις λειτουργίας.** Η λειτουργία του μικροελεγκτή γίνεται σε χαμηλές συχνότητες έτσι η κατανάλωση ισχύος κυμαίνεται σε ποσά της τάξης των mW. Επίσης μια άλλη παράμετρος που καθίστα τον μικροελεγκτή να είναι ανταγωνιστικό είναι ότι διαθέτη μετάβαση σε κατάσταση αναμονής κάτι που βοηθά την μείωση της κατανάλωσης ισχύεις.
- **Αυτονομία.** Η ενσωμάτωση συνθετών περιφερικών υποσυστημάτων που διαθέτει δεν χρειάζεται άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.

- **Επίτευξη έλεγχου ή μετρήσεων σε πραγματικό χρόνο.** Οι μικροελεγκτές δεν απαιτούν λογισμικό συγκρητηκα με H/Y για το επιτύχουν.
- **Σχεδιασμός κυκλώματος για τη λύση κάποιου προβλήματος με τη χρήση μικροελεγκτών.** Η χρήση των μικροελεγκτών για τον σχεδιασμό ηλεκτρονικών κυκλωμάτων μάς δίνει μεγάλη ευελιξία και απλοποίηση κατά τη διαδικασία αυτή, διότι ελαχιστοποιεί σε μεγάλο βαθμό τον αριθμό των εξαρτημάτων που θα περιείχε το κύκλωμα, αν χρησιμοποιούσαμε τα παραδοσιακά ψηφιακά και αναλογικά εξαρτήματα (όπως πύλες, χρονομέτρες κ.λπ.). Έτσι, με τη χρήση του κατάλληλου μικροελεγκτή και κάποιων εξαρτημάτων εισόδου (όπως φωτοαντιστάτες, διακόπτες κ.λπ.) και εξόδου (όπως βομβητών, λαμπτήρων κτλ.), μπορούμε να λύσουμε αρκετά πολύπλοκα τεχνολογικά προβλήματα, αρκεί να προγραμματίσουμε σωστά τον μικροελεγκτή.

Βεβαία υπάρχουν και κάποια μειονεκτήματα όπως:

- Δεν αλλάζει το πρόγραμμά του, αφού είναι γραμμένο στη ROM.
- Είναι δύσκολος ο προγραμματισμός του.
- Έχει μεγάλο χρόνο ανάπτυξης. Για να ολοκληρωθεί ένα προϊόν μπορεί να απαιτηθεί από 1 εβδομάδα μέχρι 1 χρόνο.

1.5 Διαδεδομένες κατηγορίες μικροελεγκτών.

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).
- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I²C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και

πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.

- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίξει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα μόντεμ.

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

1.6 Προγραμματισμός και εργαλεία ανάπτυξης.

Η επιτυχία μιας οικογένειας μικροελεγκτών καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή (assembly), προγραμματιστές της εσωτερικής μνήμης και εργαλεία εκσφαλμάτωσης (debuggers). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρείες.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

1.7 Κατασκευαστές μικροελεγκτών.

Μερικοί από τους γνωστότερους κατασκευαστές μικροελεγκτών είναι οι:

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα)
- Atmel
- Epson
- Freescale Semiconductor (πρώην Motorola)
- Hitachi
- Maxim (μετά την εξαγορά της Dallas)
- Microchip
- NEC
- Toshiba
- Texas Instruments

2.ARDUINO

2.1 Εισαγωγή στην υπολογιστική πλατφόρμα ARDUINO.

Οι πλατφόρμες Arduino κατασκευάζονται κυρίως από την εταιρία Smart Project. Ωστόσο, το Arduino ξεκίνησε ως έργο προς ανάπτυξη το 2005 στην Ιταλία, στο Ινστιτούτο Αλληλεπίδρασης Σχεδίασης στην πόλη Ιβρέα, όπου είναι η έδρα της εταιρίας Olivetti, από τους Massimo Banzi, Gianluca Martino και David Cuartielles ώστε οι φοιτητές του Ινστιτούτου να αναπτύσσουν ενσωματωμένα συστήματα οικονομικά και αποδοτικά αξιοποιώντας τις δυνατότητες και τις ευκαιρίες που μπορεί να προσφέρει το ελεύθερο λογισμικό.



Εικόνα 2.1: Gianluca Martino, Massimo Banzi, David Cuartielles δημιουργοί της πλατφόρμας Arduino

Το Arduino είναι ένας μικροελεγκτής ο οποίος περιλαμβάνει ένα chip ATmega. Με λίγα λόγια διαθέτει εισόδους και εξόδους που αντιδρούν βάση του προγραμματισμού που κάναμε και που φορτώσαμε στο chip με τη βοήθεια του υπολογιστή. Η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η Wiring, η οποία είναι αρκετά εύκολη στη σύνταξη και διατίθεται σε πλατφόρμες Linux, MAC και Windows με άδεια χρήσης GPL. Αυτό όμως που κάνει το Arduino ακόμα πιο σημαντικό είναι ότι όλο το κύκλωμα της πλακέτας διατίθεται με άδεια χρήσης Creative Commons, πράγμα που σημαίνει ότι ο καθένας μπορεί να κατασκευάσει την δική του πλακέτα όπως αυτός θέλει. Φυσικά για τους αρχάριους το να κατασκευάσουν την πλακέτα μόνοι τους, ίσως ακουστεί λίγο τραβηγμένο, οπότε είναι ευκολότερο να αγοράσουν μια έτοιμη πλακέτα Arduino από το διαδίκτυο η οποία διατίθεται σε πάρα πολύ προσιτή τιμή.

Τα παραπάνω δεν ακούγονται πρωτότυπα. Υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ποια είναι η ειδοποιός διαφορά; Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί να κατασκευαστεί από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές ακόμα και για εμπορικούς σκοπούς και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα που χρησιμοποιεί το Arduino σε

κατασκευές άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας. Γενικά, τα Projects στον εν λόγω μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP). Το Arduino χρησιμοποιεί τώρα ένα ειδικά προγραμματιζόμενο Atmega382 αντί του chip FTDI ώστε αυτό να επιτρέπει τόσο την πιο γρήγορη ταχύτητα μεταφοράς όσο και τη γρήγορη σειριακή επικοινωνία.

Βασικά πλεονεκτήματα πλατφόρμας Arduino:















- Οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών με μέγιστο κόστος 50 ευρώ.
- Τρέχει σε διάφορα Λειτουργικά Συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux όπου συγκριτικά με τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.
- Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον. Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνεται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται. Το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές.
- Ανοιχτού Υλικού το οποίο μπορεί να επεκταθεί. Ο Arduino βασίζεται στους μικροελεγκτές της Atmel ATMEGA368 και ATMEGA168. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της Creative Commons, επιτρέποντας σε έμπειρους σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσνοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα.
- Η πλακέτα arduino uno έχει τη δυνατότητα τροφοδοσίας από τη θύρα usb του υπολογιστή είτε από εξωτερικό τροφοδοτικό, είτε από μπαταρία των 9V.



Εικόνα 2.2: Arduino λογότυπο.

2.2 Εκδόσεις Arduino

Παρακάτω ακολουθούν μερικές από τις πλατφόρμες Arduino που έχουν αναπτυχθεί και όπου η κάθε μία είτε αποτελεί εξέλιξη κάποιας άλλης, είτε έχει αναπτυχθεί για κάποιο συγκεκριμένο σκοπό. Στην πτυχιακή εργασία, επιλέχθηκε το μοντέλο Arduino UNO μιας και καλύπτει όλες τις ανάγκες της εργασίας καθώς τα pins που διαθέτει επαρκούν για όλες τις λειτουργίες.

				
Arduino Uno	Arduino Leonardo	Arduino Mega 2560	Arduino LilyPad	Arduino Mega ADK
				
Arduino Fio	Arduino Ethernet	Arduino Pro	Arduino BT	Arduino Nano
				
USB/Serial Light Adapter	Arduino Mini	Mini USB/Serial Adapter	Arduino Pro Mini	

Εικόνα 2.3: Μοντέλα Arduino.

2.3 Το λογισμικό του Arduino.

Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστής και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα. Το περιβάλλον ανάπτυξης είναι βασισμένο στην Processing, ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισαγάγει στον προγραμματισμό μη εξοικειωμένους χρήστες με την ανάπτυξη λογισμικού. Η συγκεκριμένη γλώσσα προγραμματισμού προέρχεται από την Wiring, μια γλώσσα που μοιάζει με την C η οποία παρέχει παρόμοια λειτουργικότητα για μια πιο περιορισμένης σχεδίασης πλακέτα, της οποίας το περιβάλλον ανάπτυξης βασίζεται επίσης στην Processing. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino

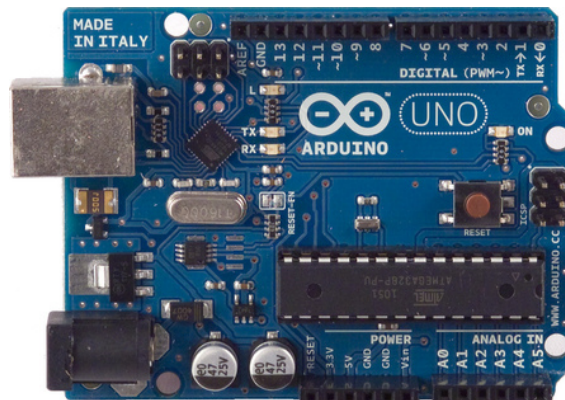
2.4 Arduino Uno

Η καρδιά του Arduino Uno είναι φυσικά ένας μικροεπεξεργαστής. Αυτός είναι το «μυαλό» του Arduino και είναι προγραμματιζόμενος ώστε να ελέγχει τα 14 ψηφιακά input/output pins και τα 6 αναλογικά που υπάρχουν πάνω στην πλακέτα ανάπτυξης. Δια μέσου αυτών των 20 pins γίνονται όλες οι διασυνδέσεις με εξωτερικά στοιχεία (κινητήρες, LEDs, LCD οθόνες κλπ) και αισθητήρες (Ultrasonic, θερμομέτρα, accelerometers κ.α).

Στην πλακέτα ανάπτυξης υπάρχει μια θύρα USB. Μέσω αυτής γίνεται η μεταφορά δεδομένων από αυτήν προς κάποια άλλη συσκευή, συνήθως έναν υπολογιστή, και το αντίστροφο. Ωστόσο, η κύρια χρήση στα αρχικά στάδια εκμάθησης είναι η μεταφορά του προγράμματος από τον υπολογιστή στον μικροεπεξεργαστή αλλά και η οπτικοποίηση των δεδομένων που απορρέουν από την λειτουργία της συσκευής μετά από το προγραμματισμό.



Εικόνα 2.4: Συσκευασία πακέτο Arduino Uno.



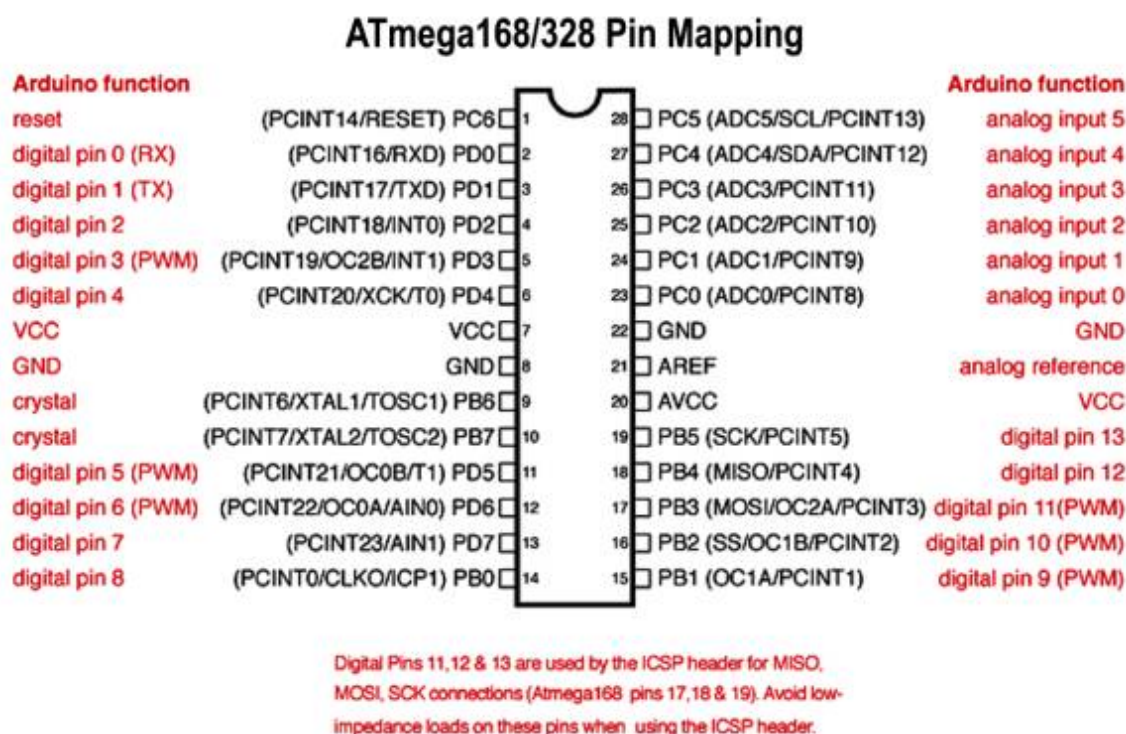
Εικόνα 2.5: Arduino Uno.

2.4.1 Χαρακτηριστικά

Στον παρακάτω πίνακα αναλύουμε τα βασικά χαρακτηριστικά του μοντέλου Arduino Uno

Μικροελεγκτής	ATMEGA328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14, (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
Μνήμη flash	32KB (ATMEGA328)
Μνήμη SRAM	2KB (ATMEGA328)
Μνήμη EEPROM	1KB (ATMEGA328)
Ταχύτητα ρολογιού	16MHz

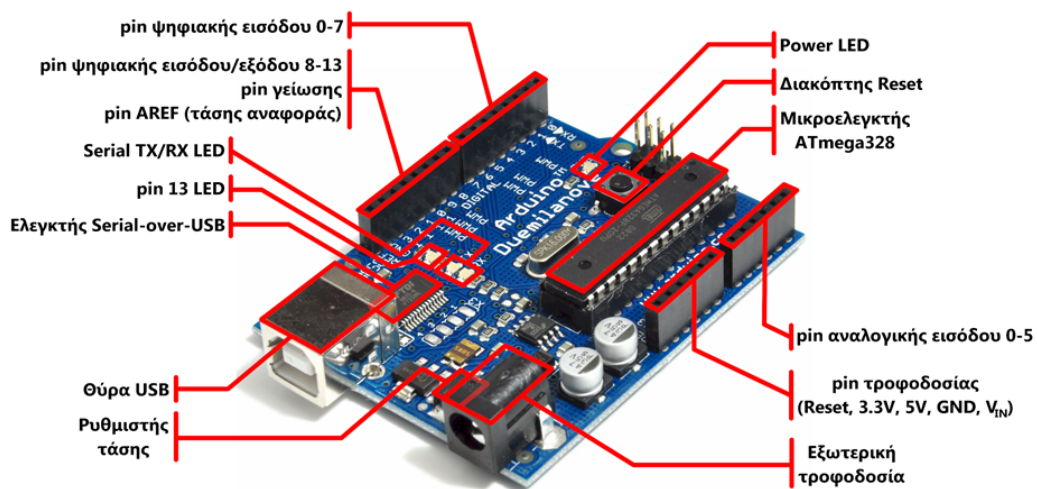
Εικόνα 2.6: Χαρακτηριστικά Arduino Uno.



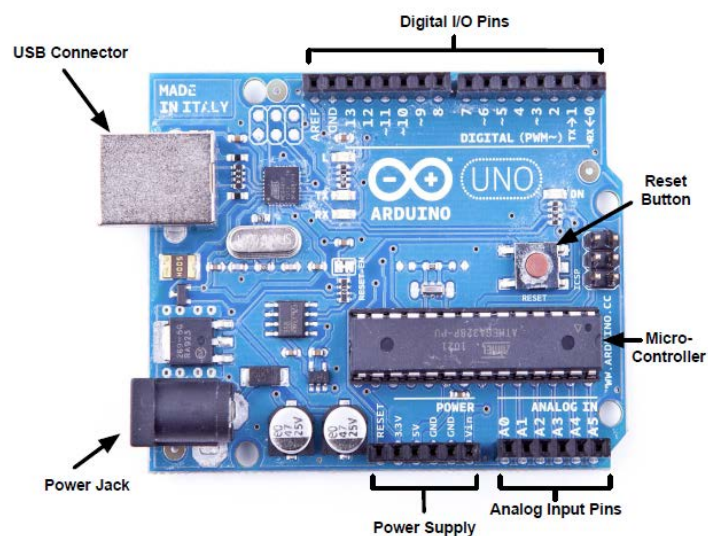
Εικόνα 2.7: Pins μικροελεγκτή ATmega 328

2.4.2 Μέρη ενός Arduino Uno

Το Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου οι οποίοι μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές-συναρτήσεις pinMode(), digitalWrite(), και digitalRead() που θα αναλυθούν παρακάτω. Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον, έχει 6 αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5 volts. Εκτός αυτού, 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως αναλογικές έξοδοι.



Εικόνα 2.8: Αναλυτικά μέρη Arduino.



Εικόνα 2.9: Γενικά μέρη Arduino.

2.4.2.1 Ακροδέκτες.

Ακροδέκτες 0 και 1: λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμα στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.

Ακροδέκτες 2 και 3: λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Ακροδέκτες 3, 5, 6, 9, 10 και 11: μπορούν να λειτουργήσουν και ως ψευδό-αναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation).

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN όπως φαίνεται και στην εικόνα 6, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με τη σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί ο ακροδέκτης AREF με 3.3V και στη συνέχεια διαβάσει κάποιον ακροδέκτη αναλογικής εισόδου στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.

2.4.2.2 Ενέργια – Τροφοδοσία

Το Arduino Uno μπορεί να λειτουργήσει μέσω τροφοδοσίας απ' την θύρα USB ή με κάποια εξωτερική πηγή ενέργειας, μπαταρία ή AC to DC adapter. Η επιλογή πηγής USB ή εξωτερικής πηγής γίνεται αυτόματα. Το βύσμα απ' τον AC to DC adapter (2.1 mm center - positive plug) μπορεί να μπει στην αντίστοιχη είσοδο ενεργείας power jack που διαθέτει το Arduino Uno, ενώ τα καλώδια της μπαταρίας μπορούν να τοποθετηθούν στα pins GND και Vin που υπάρχουν πάνω στην πλακέτα. Η τάση όπως αναφέραμε και στα βασικά χαρακτηριστικά παραπάνω μπορεί να είναι από 7 έως 12 Volt. Μπορεί βεβαίως και με λιγότερα από 7 Volt αλλά η έξοδος 5 Volt που διαθέτει μπορεί να δίνει λιγότερη τάση κάνοντας το σύστημα να υπολειτουργεί.

Οι ακροδέκτες τροφοδοσίας που διαθέτει είναι:

VIN: Πρόκειται για pin εισόδου τάσης / πηγής τροφοδοσίας, π.χ. από μπαταρία. Εάν χρησιμοποιηθεί ως έξοδος της τάσης του αντίπτορα.

5V: Το pin αυτό μας δίνει όπως λέει 5 Volt για να τροφοδοτήσουμε εξαρτήματα που έχουμε συνδέσει με το Arduino Uno. Σε αυτή την περίπτωση πρέπει να είμαστε ιδιαίτερα προσεχτική διότι ο ακροδέκτης αυτός είναι έξοδος και όχι είσοδος (VIN).

3.3V: Αντίστοιχα το pin αυτό μας δίνει 3.3 Volt και μέγιστο ρεύμα 50mA.

GND(Γείωση): Υπάρχουν συνολικά 3 pins για γείωση πάνω στην πλακέτα του Arduino Uno.

IOREF: Το συγκεκριμένο pin δίνει την τάση λειτουργίας του μικροελεγκτή και χρησιμοποιείτε κυρίως στα shields ώστε να αναγνωρίζεται από αυτά εάν το κύκλωμα δουλεύει με 5 ή 3.3 Volt.

RESET: Το συγκεκριμένο pin χρησιμοποιείτε στα shield ως επιπλέον κουμπί reset, περνώντας την τιμή λογικό 0 LOW



Εικόνα 2.10: Τρόποι τροφοδοσίας πλακέτας Arduino.

2.4.2.3 Μνήμη.

Ο μικροελεγκτής ATmega328 διαθέτει:







- 32KB μνήμη (όπου τα 0,5kB χρησιμοποιούνται από τον φορτωτή εκκίνησης (bootloader)).
- 2kB μνήμη τυχαίας προσπέλασης τύπου SRAM η οποία χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται.
- 1kB μνήμη τύπου EEPROM στην οποία αποθηκεύονται οι τιμές των μεταβλητών όταν η πλατφόρμα σβήσει(OFF). Χρησιμοποιείται για την αποθήκευση ρυθμίσεων και άλλων παραμέτρων ανάμεσα στα Reset του Arduino.

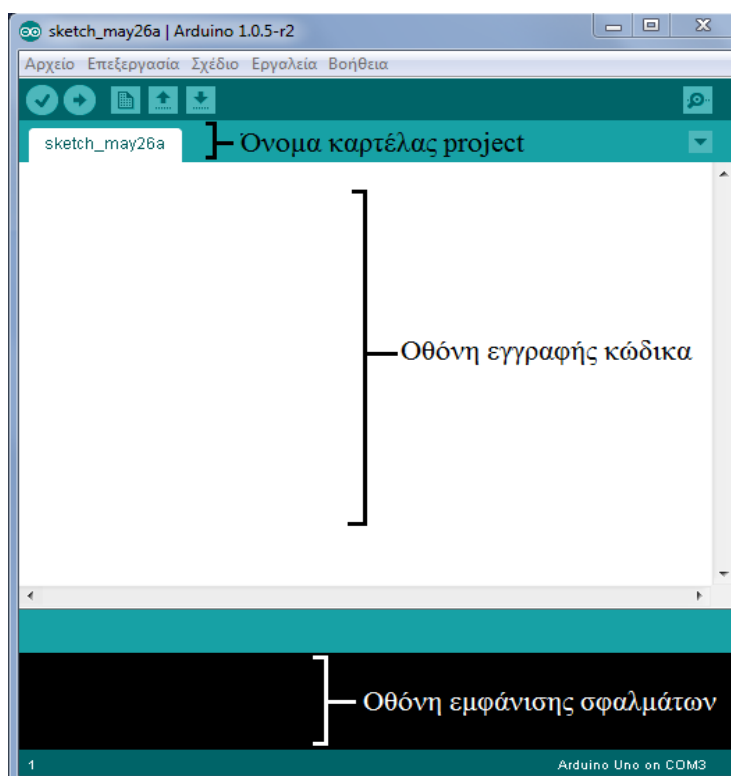
2.4.2.4 Προγραμματισμός.

Το Arduino μπορεί να προγραμματιστεί με χρήση του προγράμματος που παρέχει η εταιρία κατασκευής του. Ο μικροελεγκτής ATmega328 έχει προεγκατεστημένο ένα πρόγραμμα εκκίνησης που επιτρέπει να φορτώνονται κώδικες στην πλακέτα χωρίς να χρειάζεται εξωτερική πλακέτα προγραμματισμού. Η επικοινωνία με τον μικροελεγκτή γίνεται με χρήση του πρωτοκόλλου STK500.

2.4.3 Περιβάλλον ανάπτυξης Arduino IDE.

Όπως αναφέραμε το Arduino και πιο συγκεκριμένα το Arduino Uno με το οποίο θα ασχοληθούμε, είναι ένα υπολογιστικό σύστημα. Έτσι για να μπορέσει να εκτελέσει διαφορές λειτουργίες θα πρέπει να το προγραμματίσουμε. Το Arduino IDE είναι το περιβάλλον ανάπτυξης ή αλλιώς προγραμματισμού του Arduino. Το Arduino IDE περιέχει ένα πρόγραμμα επεξεργασίας κειμένου, για τη σύνταξη του κώδικα, μια περιοχή στην οποία εμφανίζονται μηνύματα, μία κονσόλα κειμένου και μια γραμμή εργαλείων υπό μορφή κουμπιών. Συνδέεται με το hardware μέρος του arduino για να φορτώσει προγράμματα και να επικοινωνεί μαζί τους. Ο κώδικας που έχει γραφεί για το Arduino ονομάζεται sketch. Παρακάτω παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης, υπό μορφή κουμπιών και στην εικόνα το ίδιο το περιβάλλον.

-  Ελέγχει εάν υπάρχουν σφάλματα στον κώδικα μας
-  Περνάει το πρόγραμμα στο Arduino
-  Ανοίγει ένα νέο κενό project
-  Άνοιγμα αρχείου
-  Αποθήκευση αρχείου
-  Άνοιγμα σειριακής οθόνης



Εικόνα 2.11: Περιβάλλον Arduino IDE.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση.
- Μερικές έτοιμες βιβλιοθήκες για προέκταση της.
- Τον compiler για τη μεταγλώττιση των sketch.
- Μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση των sketch.
- Την επιλογή για ανέβασμα των μεταγλωτισμένων sketch στο Arduino.

Γλώσσα: Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++.

Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από τη C, στη γλώσσα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino.

Τα προγράμματα του Arduino διαίρούνται σε τρία μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions).

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.

<Δήλωση μεταβλητών>

```
void setup(){
```

<Δήλωση λειτουργιών που ισχύουν για όλο το πρόγραμμα>

```
}
```

```
void loop(){
```

<Δήλωση των λειτουργιών που επαναλαμβάνονται κατά τη λειτουργία του προγράμματος>

```
}
```

Οι πιο σημαντικές εντολές ανά μέρη επεξηγούνται στον πίνακα που ακολουθεί:

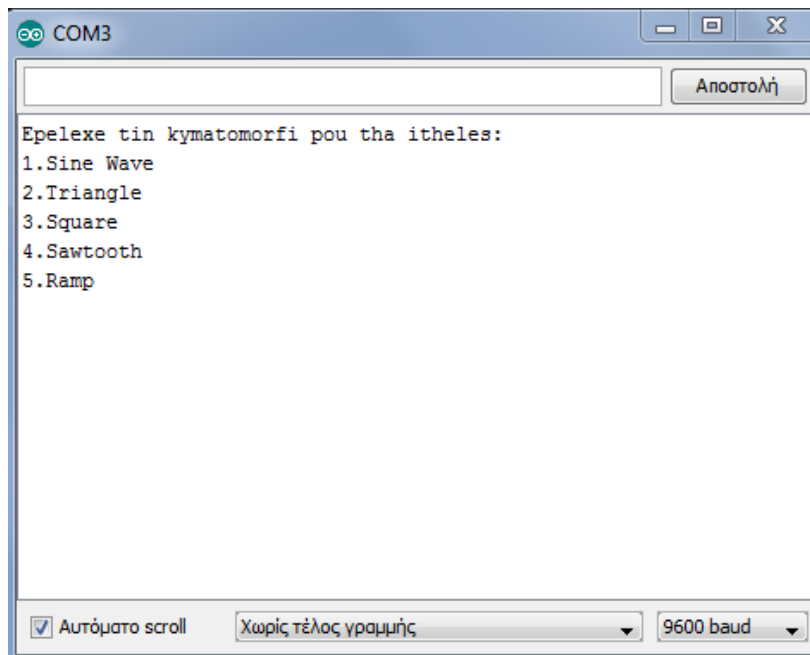
ΔΟΜΗ	ΤΙΜΕΣ	ΣΥΝΑΡΤΗΣΕΙΣ
setup()	Σταθερές	Ψηφιακές I/O
loop()	HIGH – LOW	pinMode()
Έλεγχος	INPUT – OUTPUT	digitalWire()
if	true - false	digitalRead()
if..else	integer constants	Αναλογικές I/O
for	floating point constants	analogReference()
switch case	Τύποι δεδομένων	analogRead()
while	void	analogWrite()
do...while	boolean	Ειδικές I/O
break	(unsigned)char	tone(), noTone()
continue	byte	shiftOut()
return	(unsigned)int	pulseIn()
goto	word	Χρόνος
Σύνταξη	(unsigned)long	millis(), micros()
;	float	delay(), delayMicroseconds()
{ }	double	Μαθηματικές
// ή /* */	string (πίνακας char)	min(), max(), abs(), constraint(), map()
#define	String (αντικείμενο)	pow(), sqrt(), sin(), cos(), tan()
#include	array	randomSeed(), random()
Αριθμητικοί τελεστές	Μετατροπή	Bits/Bytes
=	char(), byte(), int(), word()	lowByte(), highByte(), bitRate(), bitWrite
+, -, *, /, %	long(), float()	bitSet(), bitClear(), bit()
Σύγκριση	Πεδίο μεταβλητών	Interrupts
==, !=	variable scope, static	attachInterrupt(), detachInterrupt()
<, >, <=, >=	volatile, const	interrupts(), noInterrupts()
Λογικοί τελεστές	Εργαλεία	Επικοινωνία
&&, , !	sizeof()	Serial

Πίνακας.(1): Γενικές εντολές της γλώσσας του Arduino

2.4.4 Σειριακή οθόνη

Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino. Πιο συγκεκριμένα, η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί send ή πατώντας το Enter. Επίσης, στο κάτω μέρος της σειριακής οθόνης, μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (baud) από την λίστα που

εμφανίζεται ανάλογα με την τιμή που θα επιλεγθεί στο προγραμματισμό του Arduino με το Serial.begin().



Εικόνα 2.12: Serial monitor Arduino IDE

2.4.5 Εγκατάσταση προγράμματος του μικροελεγκτή Arduino.

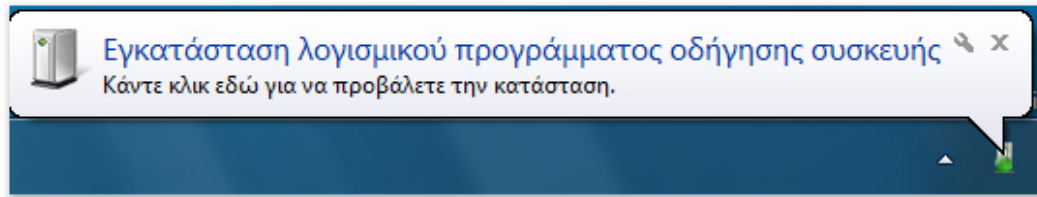
Για να γίνει σωστή εγκατάσταση του προγράμματος, πρέπει να ακολουθηθεί μια σειρά από βήματα, ανάλογα με το λειτουργικό σύστημα που διαθέτει. Στην περίπτωση μας θα εγκατασταθεί σε λειτουργικό σύστημα των Windows 7.

1. Συνδέουμε τον μικροελεγκτή Arduino με τον ηλεκτρονικό υπολογιστή μας μέσω καλωδίου τύπου USB.



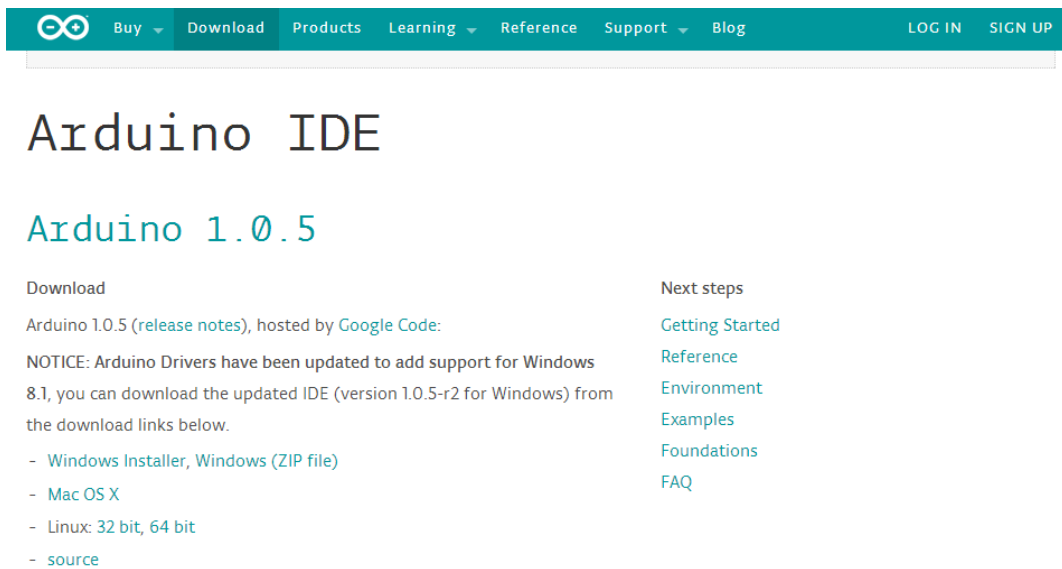
Εικόνα 2.13: Arduino UNO και καλώδιο σύνδεσης με H/Y.

2. Στα Windows 7 κάνει αυτόματη εγκατάσταση των οδηγών του μικροελεγκτή.



Εικόνα 2.14: Windows 7 αυτόματη εγκατάσταση συσκευών.

3. Μεταφορτώνουμε δωρεάν την τελευταία έκδοση του Arduino στην ηλεκτρονική σελίδα της εταιρίας <http://arduino.cc/en/Main/Software>.



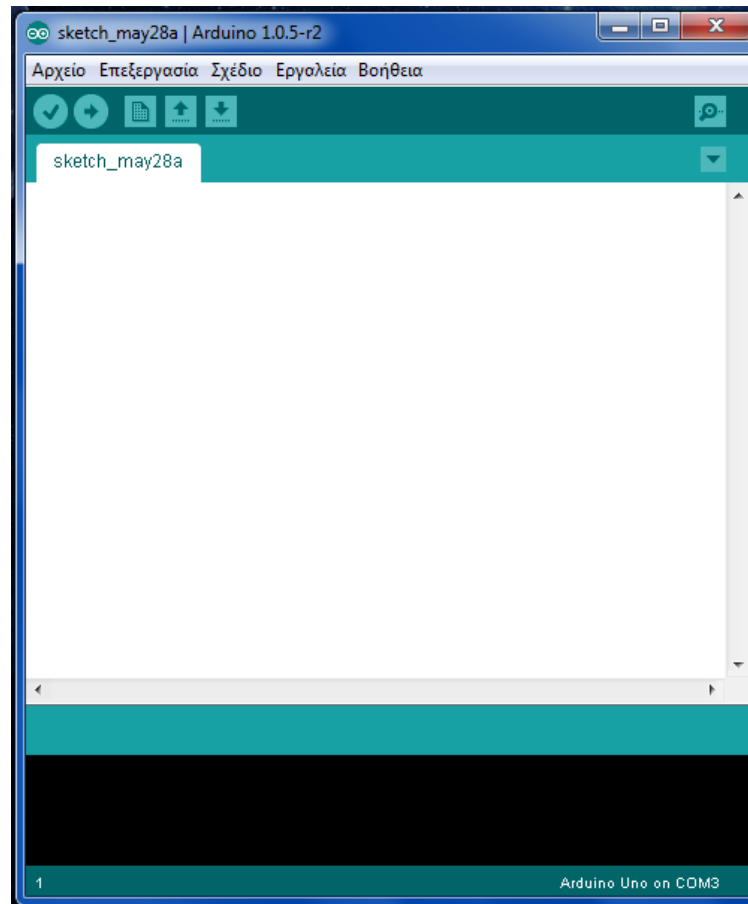
Εικόνα 2.15: Κεντρική σελίδα για μεταφόρτωση έκδοσης του Arduino.

4. Αφού κάνουμε εγκατάσταση την τελευταία έκδοση στην επιφάνια εργασίας θα βρούμε το εικονίδιο της εφαρμογής.



Εικόνα 2.16: Εικονίδιο εφαρμογής.

5. Κάνουμε διπλό αριστερό κλικ με το ποντική επάνω στο εικονίδιο για να ανοίξουμε την εφαρμογή.



Εικόνα 2.17: Περιβάλλον ανάπτυξης Arduino.

6. Από το περιβάλλον ανάπτυξης του, από το μενού Εργαλεία επιλέγουμε για Πλακέτα το Arduino Uno και για Σειριακή θύρα το COM3.

3. ΥΛΟΠΟΙΗΣΗ ΚΑΤΑΣΚΕΥΗΣ.

3.1 Γενική περιγραφή γεννήτριας συχνοτήτων και παλμογράφου.

Πριν αρχίσουμε να αναλύουμε βήμα την κατασκευή της γεννήτριας συχνοτήτων με την χρήση του μικροελεγκτή Arduino είναι αναγκαίο να αναφέρουμε την ιδιότητα μιας γεννήτριας συχνοτήτων καθώς και την ιδιότητα του παλμογράφου που θα μας βοηθήσει να απεικονίσουμε τα σήματα που θα παράγουμε μέσω της γεννήτριας.

3.1.1 Γεννήτρια συχνοτήτων.

Οι γεννήτριες συχνοτήτων είναι συσκευές που μπορούν να δώσουν σήματα ημιτονικά, τριγωνικά και τετραγωνικά μεταβλητού πλάτους και συχνότητας. Σε κάθε γεννήτρια υπάρχουν δυο ακροδέκτες για τη λήψη του σήματος. Ο ένας έχει το σύμβολο της γείωσης και πρέπει να συνδέεται πάντα (εκτός μερικών περιπτώσεων) στο κοινό σημείο του κυκλώματος μέσω ενός μαύρου καλωδίου. Ο άλλος ακροδέκτης συνδέεται πάντα μέσω ενός κόκκινου καλωδίου στο σημείο του κυκλώματος που θα δώσουμε το σήμα.



Εικόνα 3.1 Γεννήτρια συχνοτήτων.

3.1.2 Παλμογράφος.

Ο παλμογράφος είναι το πλέον διαδεδομένο ηλεκτρονικό όργανο μέτρησης, το οποίο παρέχει τη δυνατότητα απεικόνισης σε μια οθόνη της κυματομορφής ενός σήματος τάσης, με συχνότητα από το μηδέν (συνεχής τάση) έως και αρκετά GHz. Συνήθως η απεικόνιση του μετρούμενου μεγέθους γίνεται ως προς το χρόνο $u(t)$, αν και υπάρχει η δυνατότητα

παρουσίασης στην οθόνη της τάσης $u_1(t)$ ως προς μια άλλη τάση $u_2(t)$, δηλαδή του μεγέθους $u_1(u_2)$. Σημειώνουμε ότι ο παλμογράφος μπορεί να μετρήσει απευθείας μόνο διαφορά δυναμικού.

Οι παλμογράφοι διακρίνονται σε δύο κύριες κατηγορίες:

- Απλοί αναλογικοί παλμογράφοι.
- Παλμογράφοι με μνήμη.

Οι απλοί αναλογικοί παλμογράφοι μπορούν να μετρήσουν μόνο περιοδικά σήματα. Επιπλέον, η συχνότητα του μεταβαλλόμενου σήματος πρέπει να είναι μεγαλύτερη από 30Hz περίπου, έτσι ώστε η απεικόνιση του σήματος να ανανεώνεται επαρκώς γρήγορα στην οθόνη και να φαίνεται σταθερή στο μάτι. Οι παλμογράφοι με μνήμη δεν έχουν τους παραπάνω περιορισμούς και μπορούν να καταγράψουν φαινόμενα που εκδηλώνονται μόνο μια φορά.



Εικόνα 3.2: Αριστερά απεικονίζεται ο απλός παλμογράφος και δεξιά ο παλμογράφος με μνήμη.

3.2 Συστατικά μέρη κατασκευής.

Τα συστατικά της κατασκευής αποτελούνται από υλικά μέρη, την μελέτη παράγωγης σημάτων που θα ενσωματώσουμε στον κώδικα των κώδικα των όποιο θα δημιουργήσουμε και θα προγραμματίσουμε τον μικροελεγκτή Arduino που παίζει κύριος και τον σημαντικότερο ρολό και θα αναλύσουμε στο επόμενο κεφαλαίο.

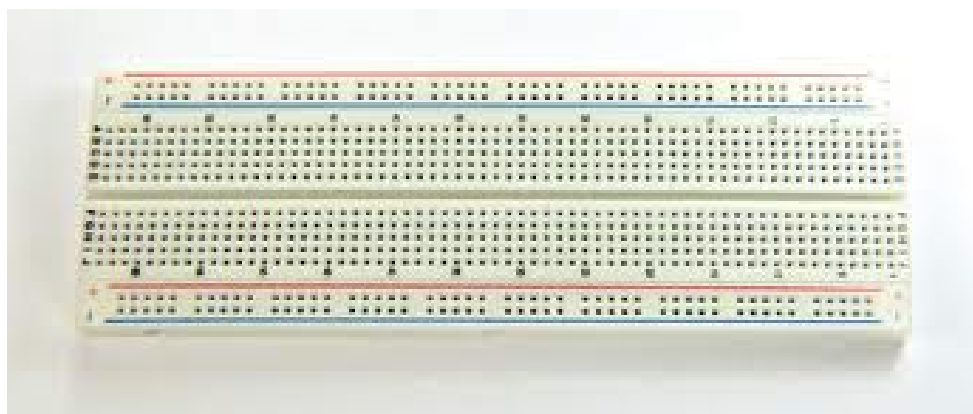
Τα υλικά μέρη είναι:

Μικροελεγκτής Arduino UNO: Πρόκειται για την “κάρδια” της γεννήτριας αφού επάνω του θα συνδεθεί το κύκλωμα και θα προγραμματιστεί για να γίνει η παραγωγή σημάτων τον όποιον η απεικόνιση θα γίνει μέσω του παλμογράφου.

BreadBroad (raster): Το ράστερ αποτελεί βασικό και αναγκαίο εργαλείο για την προσωρινή κατασκευή του κυκλώματος. Το ράστερ είναι μια πλακέτα με πλαστικό εξωτερικό περίβλημα και διαθέτει αρκετά σημεία επαφής τα οποία απέχουν μεταξύ τους συνήθως 2.54mm. Επίσης, σε αυτά τα σημεία επαφής μπορούν να τερματιστούν τόσο απλά καλώδια όσο και διάφορα ηλεκτρονικά εξαρτήματα όπως για παράδειγμα αντιστάτες, πυκνωτές, ολοκληρωμένα κυκλώματα.

Αξίζει να αναφερθεί, πως τα διάφορα σημεία επαφής που υπάρχουν σε ένα ράστερ είναι οργανωμένα οριζόντια και κάθετα σε λωρίδες. Αυτές οι λωρίδες διακρίνονται είτε σε λωρίδες

τροφοδοσίας, είτε σε λωρίδες τερματισμού. Επίσης, κάθε ράστερ διαθέτει τουλάχιστον μία ή περισσότερες λωρίδες τερματισμού καθώς και μία ή περισσότερες λωρίδες τροφοδοσίας



Εικόνα 3.3: BreadBroad (raster).

Πυκνωτής: Ο πυκνωτής είναι ένα εξάρτημα που χρησιμεύει ως αποθήκη ηλεκτρικού φορτίου και επομένως ηλεκτρικής ενέργειας. Αποτελείται από δύο αγωγούς που διαχωρίζονται από ένα μονωτικό υλικό. Μονάδα μέτρησης της χωρητικότητας του πυκνωτή είναι το 1 Φαράντ *Farad* (F). Πρόκειται όμως για μεγάλη μονάδα, που σπάνια χρησιμοποιείται στην πράξη. Συνήθως χρησιμοποιούνται τα υποπολλαπλάσια του μικροφαράντ (μF), νανοφαράντ (nF) και πικοφαράντ (pF). Στην δική μας περίπτωση χρησιμοποιούμε έναν πυκνωτή 1 μF .



Εικόνα 3.4: Πυκνωτής.

Ηλεκτρική αντίσταση: Η αντίσταση γενικά, όπως το λέει και η λέξη, μειώνει την τάση φέρνοντας αντίσταση, όταν περνάει από μέσα της η τάση αυτή. Με αποτέλεσμα, να έχουμε την επιθυμητή τάση στο κύκλωμά μας. Αυτό επιτυγχάνεται, ξεχωριστά για κάθε κύκλωμα, η χρήση της σωστής αντίστασης: υπάρχουν πολλών ειδών που άλλες μειώνουν την τάση λιγότερο, κι άλλες περισσότερο. Για να καταλάβουμε ποσά ohms είναι κάθε αντίσταση που έχουμε στα χεριά μας αρκεί να διακρίνουμε την χρωματική διάταξη της αντίστασης.

Για παράδειγμα έχω μια αντίσταση (Εικόνα.):

Η πρώτη γραμμούλα είναι κόκκινη που σημαίνει 2.

Η δεύτερη επίσης κόκκινη, άρα πάλι 2.

Η τρίτη είναι κίτρινη, που σημαίνει ότι είναι 4 μηδενικά.

Η τέταρτη, είναι χρυσή που σημαίνει ότι υπάρχει απόκλιση 5%.

(η ασημί δίνει 10%, η καφέ 1%, η κόκκινη 2%)

Άρα η αντίστασή μας είναι 220000 Ohms ή 220k½

Λόγω της 5% απόκλισης, θα μας δώσει από 209000 ως 231000 Ohms.



Εικόνα 3.5: Αντίσταση παραδείγματος.

0 = μαύρο

2 = κόκκινο

4 = κίτρινο

6 = μπλε

8 = γκρι

1 = καφέ

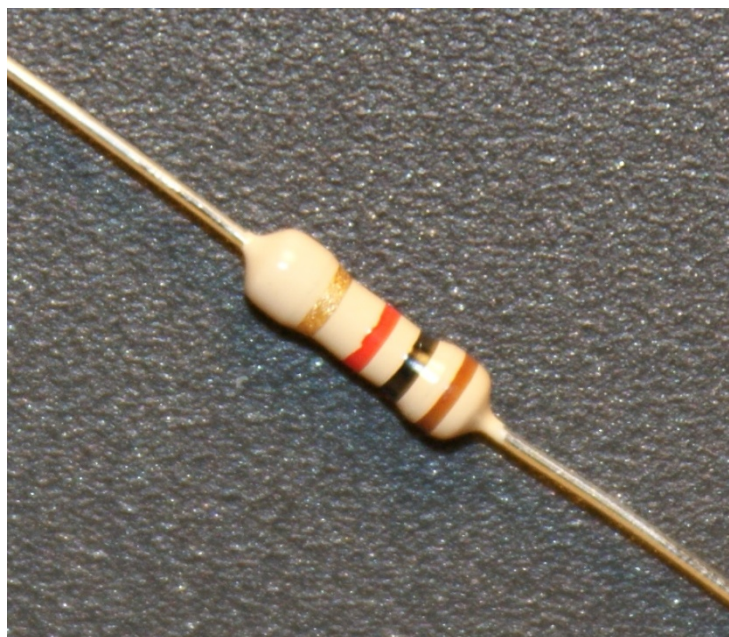
3 = πορτοκαλί

5 = πράσινο

7 = βιολετί

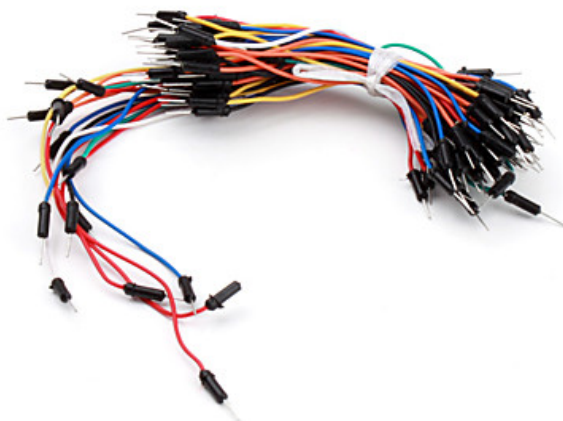
9 = άσπρο

Τώρα για παράδειγμα αν έχει πέντε μπάρες με χρώματα, οι πρώτες 3 δείχνουν νούμερα, η 4η δείχνει τα μηδενικά, η 5η δείχνει την απόκλιση. Για το δικό μας κύκλωμα θα χρησιμοποιήσουμε μια αντίσταση των 10K ohm.



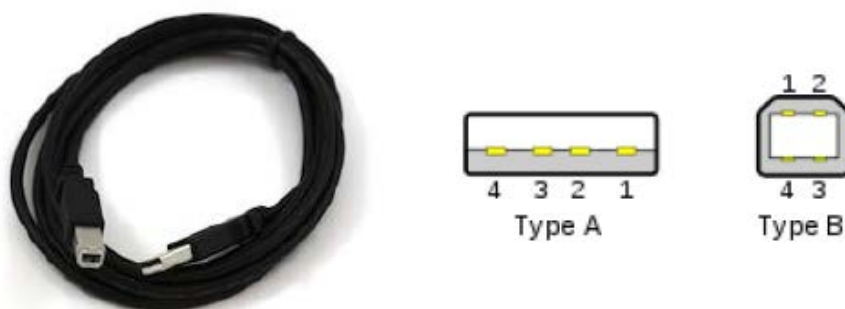
Εικόνα 3.6: Αντίσταση 10K ohm.

Καλώδια: Τα καλώδια θα μας χρησιμεύσουν στην σύνδεση μεταξύ των διεπαφών του μικροελεγκτή, με τις άκρες των εξαρτημάτων που είναι τοποθετημένα στο ράστερ και το ίδιο το ράστερ. Για ευκολία μπορούμε να προμηθευτούμε έτοιμα καλώδια, διαφορετικών μηκών και χρωμάτων που διαθέτουν στα άκρα τους εκλεπτυσμένους ακροδέκτες για την εύκολη σύνδεση τους.



Εικόνα 3.7: Καλώδια.

USB καλώδια πλατφόρμων Arduino: Επίσης απαραίτητη είναι η χρήση ειδικού καλωδίου USB για τον προγραμματισμό και την τροφοδοσία των πλατφορμών Arduino. Πιο συγκεκριμένα, χρησιμοποιήθηκε USB 2.0 τύπου A-B (αρσενικό σε αρσενικό) για σειριακή επικοινωνία. Στις τηλεπικοινωνίες, αλλά και στην επιστήμη των υπολογιστών γενικότερα, σειριακή επικοινωνία (serial communication) είναι η διαδικασία αποστολής δεδομένων ενός bit κάθε φορά, διαδοχικά μέσω ενός καναλιού επικοινωνίας.



Εικόνα 3.8: Καλώδιο USB και απεικόνιση ακροδεκτών.

3.3 Κόστος κατασκευής.

Το κόστος της κατασκευής αναγράφεται αναλυτικά στον παρακάτω πίνακα.

Εξάρτημα	Τιμή (με ΦΠΑ)
Μικροελεγκτής Arduino UNO	28.06 €
BreadBroad (raster)	5.23 €
Πυκνωτής	0.32 €
Ηλεκτρική αντίσταση	0.30 €
Καλώδια	6.73 €
USB καλώδια πλατφόρμων Arduino	2.34 €
Σύνολο	42.98 €

Πίνακας.(2): Κόστος κατασκευής.

Η αγορά έγινε από το ηλεκτρονικό κατάστημα www.robotstore.gr. Η αποστολή διήρκεσε περίπου 15 εργάσιμες ημέρες. Στο συνολικό κόστος θα προσθέσουμε επιπλέον 10 ευρώ για τα μεταφορικά της ταχυδρομικής εταιρίας.

3.4 Μελέτη και σχεδίαση σημάτων.

Αφού είδαμε στα προηγούμενα υποκεφάλαια του 3^{ου} κεφαλαίου από ποια εξαρτήματα αποτελείται η κατασκευή μας μένει να εξετάσουμε την δημιουργία των σημάτων που θα μεταφορτώσουμε σε μορφή κώδικα για την ενσωμάτωση στον μικροελεγκτή και αργότερα η απεικόνιση μέσω παλμογράφου.

Τα σήματα τα όποια θα δημιουργήσουμε είναι τα εξής:

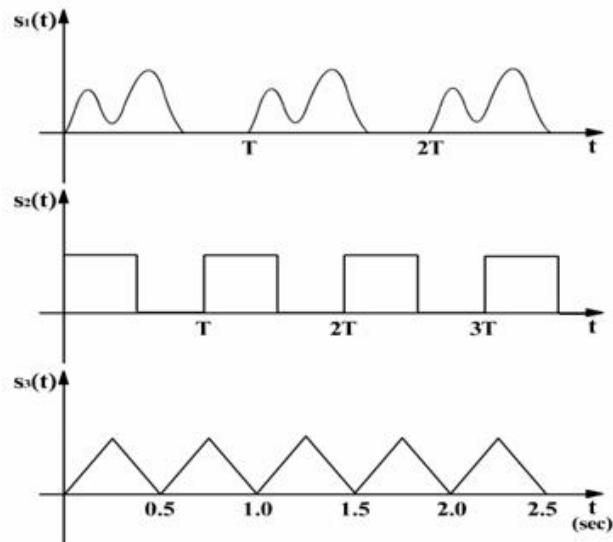
- Sine wave (Ημίτονο).
- Square wave (Τετραγωνικό σήμα).
- Triangle wave (Τριγωνικό σήμα).
- Sawtooth wave (Πριονωτό σήμα).
- Ramp wave (Κυματομορφή Ράμπας).

Γενικά

Για να διευκολύνουμε τη μελέτη των σημάτων στην ηλεκτρονική και ιδιαίτερα στις τηλεπικοινωνίες, τα διακρίνουμε σε:

- Περιοδικά σήματα και
- μη περιοδικά σήματα.

Ένα σήμα το αποκαλούμε περιοδικό, όταν επαναλαμβάνεται στο χρόνο, δηλαδή παίρνει μετά από συγκεκριμένο χρόνο τις ίδιες τιμές.



Εικόνα 3.9: Περιοδικά σήματα.

Αυτός ο συγκεκριμένος χρόνος ονομάζεται **περίοδος** και συμβολίζεται T .

Δηλαδή, $s(t) = s(t+nT)$, όπου n ακέραιος αριθμός

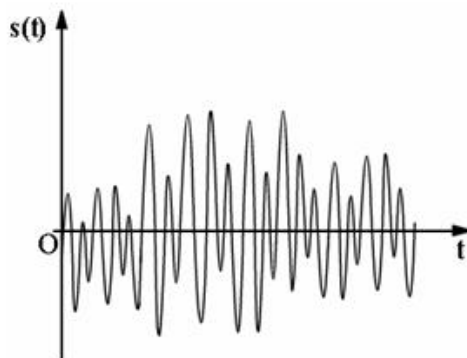
Στο παράδειγμα του σχήματος η περίοδος του σήματος είναι $T = 0,5\text{sec} = 500\text{msec}$.

Το αντίστροφο της περιόδου ενός περιοδικού σήματος ονομάζεται **συχνότητα**, συμβολίζεται F (ή f) και μετριέται σε Ερτζ (Hertz = sec^{-1} : Hz).

Για το παράδειγμα μας ισχύει:

$$F = 1/T = 1/(5 \cdot 10^{-1}) = 2 \text{ sec}^{-1} = 2 \text{ Hz}.$$

Τα τυχαία σήματα που δεν παρουσιάζουν επαναληπτικότητα ονομάζονται μη περιοδικά σήματα. Για παράδειγμα η ομιλία μας είναι μη περιοδικό σήμα, το σήμα που θα προκύψει στην έξοδο του φωτοκύτταρου από την σάρωση μιας εικόνας με την κάμερα είναι μη περιοδικό σήμα.



Εικόνα 3.10: Μη περιοδικό σήμα.

Τάση.

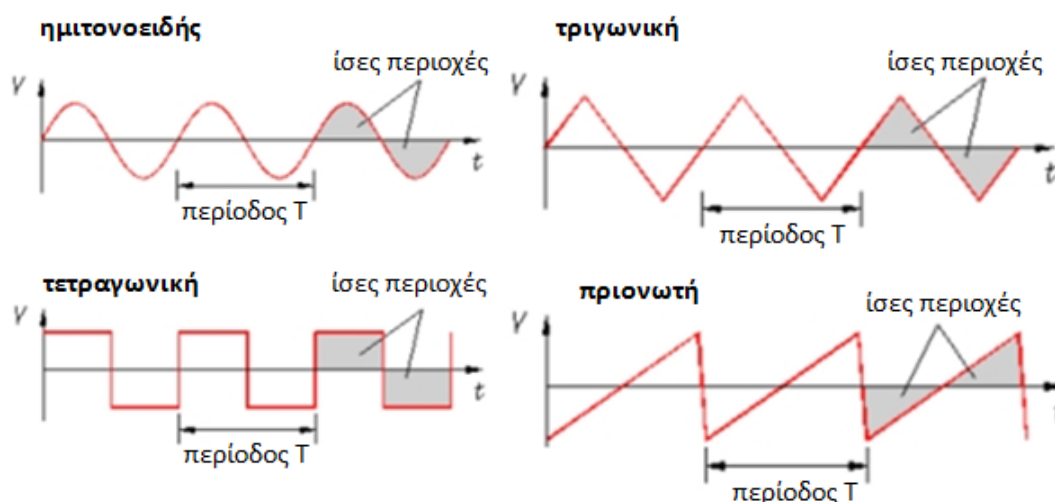
Η τάση είναι ένα μέτρο της διαφοράς του ηλεκτρικού δυναμικού μεταξύ δύο σημείων και καθορίζεται από την ηλεκτρική πηγή.

Η γραφική παράσταση της τάσης συναρτήσει του χρόνου είναι μία χρήσιμη μέθοδος για να περιγράφονται οι αλλαγές που συμβαίνουν στα ηλεκτρονικά κυκλώματα.

Όταν η τιμή της τάσης παραμένει σταθερή σε συνάρτηση με τον χρόνο, τότε ονομάζεται συνεχής ή DC τάση. Τα σήματα που αποτελούνται από επαναλαμβανόμενα σχήματα ονομάζονται κυματομορφές.

Τα σήματα που εναλλάσσουν τα επίπεδα τάσης μεταξύ θετικών και αρνητικών τιμών ονομάζονται εναλλασσόμενα ή AC σήματα.

Οι εναλλασσόμενες κυματομορφές έχουν ως χαρακτηριστικό τις ίσες περιοχές πάνω και κάτω από τον άξονα των X. Ανάλογα με το σήμα οι κυματομορφές διακρίνονται σε ημιτονοειδείς, τετραγωνικές, τριγωνικές και πριονωτές.



Εικόνα 3.11: Εναλλασσόμενες κυματομορφές.

3.4.1 Ημίτονο.

Το απλούστερο περιοδικό σήμα είναι το ημιτονικό σήμα, που το έχουμε γνωρίσει και από το μάθημα της γεωμετρίας. Είναι το βασικότερο σήμα στις τηλεπικοινωνίες και το χρησιμοποιούμε πάντοτε σαν σήμα αναφοράς, για να περιγράψουμε και να χαρακτηρίσουμε όλα τα άλλα ηλεκτρικά σήματα.

Υπενθυμίζουμε ότι το ημιτονικό σήμα ονομάζεται έτσι, διότι οι τιμές που παίρνει σε μία περίοδο T αντιστοιχούν στις τιμές του ημιτόνου της γωνίας που σχηματίζεται στον τριγωνομετρικό κύκλο, όταν ένα κινητό, ξεκινώντας από το σημείο O , κινείται για να

διαγράψει πλήρη κύκλο στο χρόνο μιάς περιόδου με σταθερή γωνιακή ταχύτητα. Η πλήρης γωνία του τριγωνομετρικού κύκλου είναι 2π .

Για την δημιουργία του ημιτόνου όπως και για τις άλλες κυματομορφές θα χρησιμοποιήσουμε ένα σύνολο 40 τιμών – σημείων.

Για όλες τις κυματομορφές μπορώ να πολλαπλασιάσω με $(39/40)$ για να βγει ακριβώς στο σημείο ώστε να μην έχω ασυνέχεια φάσης.

Επίσης κάθε υπολογισμός έγινε με την βοήθεια των λογιστικών φύλλων Excel.

$$\text{Άρα } 2\pi/40 = \pi/20, \quad \pi/20 * (0,1,2,\dots,39) \leftarrow \text{Σχέση (1)}$$

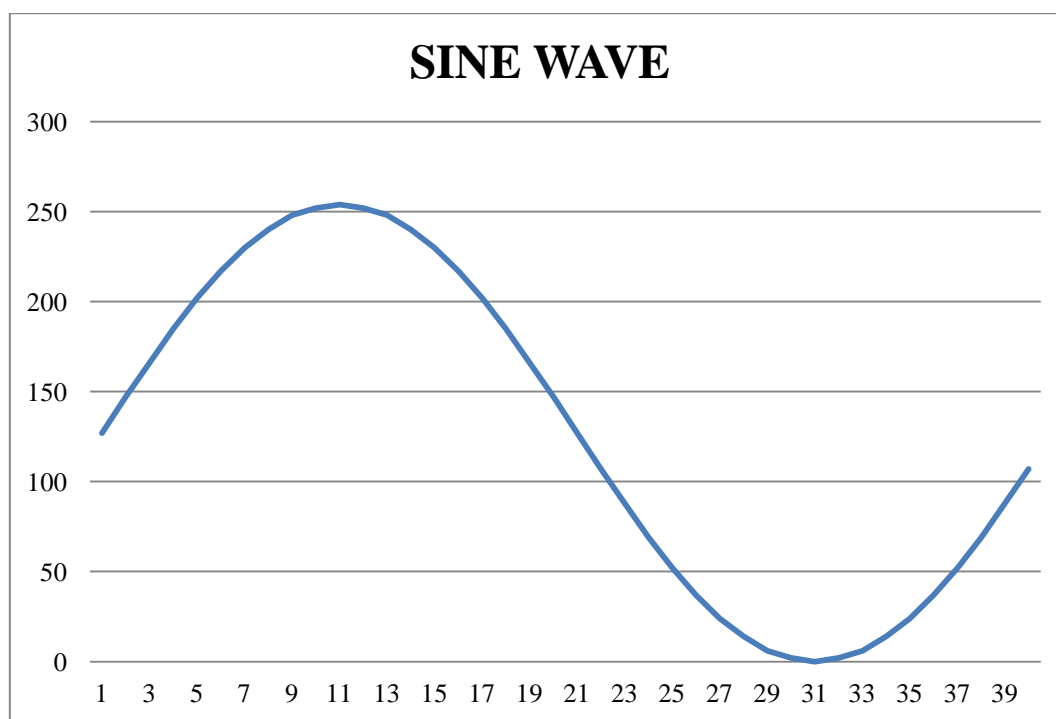
Σύνολο μεγίστου 255 της περιόδου.

$$\text{Ημιπερίοδος } 255/2=127$$

$$\text{Sin (Σχέση (1))*127} \leftarrow \text{Σχέση (2)}$$

$$\text{Σχέση (2)} + 127$$

Και τέλος το αποτέλεσμα που θα βρούμε για κάθε τιμή το στρογγυλοποιούμε.



Εικόνα 3.12: Απεικόνιση ημιτόνου.

Πιο αναλυτικά η τιμές αναφέρονται στον παρακάτω πίνακα.

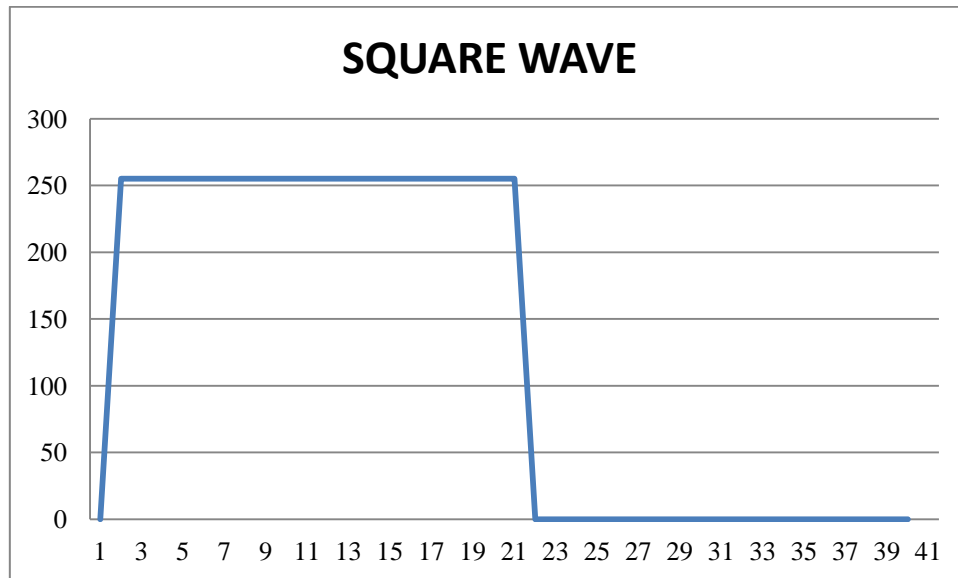
Τιμές	Φάση κάθε $2\pi/40$	Sin	Sin+127	Σημεία
0	0	0	127	127
1	0,157079633	19,86718	146,8672	147
2	0,314159265	39,24516	166,2452	166
3	0,471238898	57,65679	184,6568	185
4	0,628318531	74,64873	201,6487	202
5	0,785398163	89,80256	216,8026	217
6	0,942477796	102,7452	229,7452	230
7	1,099557429	113,1578	240,1578	240
8	1,256637061	120,7842	247,7842	248
9	1,413716694	125,4364	252,4364	252
10	1,570796327	127	254	254
11	1,727875959	125,4364	252,4364	252
12	1,884955592	120,7842	247,7842	248
13	2,042035225	113,1578	240,1578	240
14	2,199114858	102,7452	229,7452	230
15	2,35619449	89,80256	216,8026	217
16	2,513274123	74,64873	201,6487	202
17	2,670353756	57,65679	184,6568	185
18	2,827433388	39,24516	166,2452	166
19	2,984513021	19,86718	146,8672	147
20	3,141592654	1,56E-14	127	127
21	3,298672286	-19,8672	107,1328	107
22	3,455751919	-39,2452	87,75484	88
23	3,612831552	-57,6568	69,34321	69
24	3,769911184	-74,6487	52,35127	52
25	3,926990817	-89,8026	37,19744	37
26	4,08407045	-102,745	24,25484	24
27	4,241150082	-113,158	13,84217	14
28	4,398229715	-120,784	6,215822	6
29	4,555309348	-125,436	1,563581	2
30	4,71238898	-127	0	0
31	4,869468613	-125,436	1,563581	2
32	5,026548246	-120,784	6,215822	6
33	5,183627878	-113,158	13,84217	14
34	5,340707511	-102,745	24,25484	24
35	5,497787144	-89,8026	37,19744	37
36	5,654866776	-74,6487	52,35127	52
37	5,811946409	-57,6568	69,34321	69
38	5,969026042	-39,2452	87,75484	88
39	6,126105675	-19,8672	107,1328	107

Πίνακας.(3): Τιμές ημιτόνου.

3.4.2 Τετραγωνικό σήμα.

Η δημιουργία του τετραγωνικού σήματος ή του τετραγωνικού παλμού όπως συνηθίζουμε να λέμε είναι πιο απλή από κάθε άλλη κυματομορφή.

Στην πρώτη ημιπερίοδο δηλαδή τα 20 πρώτα σημεία 255 και τα επόμενα 0.



Εικόνα 3.13: Απεικόνιση τετραγωνικού παλμού.

Πιο αναλυτικά η τιμές αναφέρονται στον παρακάτω πίνακα.

Τιμές	Φάση κάθε 2π/40	Σημεία
0	0	0
1	0,157079633	255
2	0,314159265	255
3	0,471238898	255
4	0,628318531	255
5	0,785398163	255
6	0,942477796	255
7	1,099557429	255
8	1,256637061	255
9	1,413716694	255
10	1,570796327	255
11	1,727875959	255
12	1,884955592	255
13	2,042035225	255
14	2,199114858	255
15	2,35619449	255
16	2,513274123	255
17	2,670353756	255
18	2,827433388	255
19	2,984513021	255
20	3,141592654	255
21	3,298672286	0
22	3,455751919	0
23	3,612831552	0
24	3,769911184	0
25	3,926990817	0
26	4,08407045	0
27	4,241150082	0
28	4,398229715	0
29	4,555309348	0
30	4,71238898	0
31	4,869468613	0
32	5,026548246	0
33	5,183627878	0
34	5,340707511	0
35	5,497787144	0
36	5,654866776	0
37	5,811946409	0
38	5,969026042	0
39	6,126105675	0

Πίνακας.(4): Τιμές τετραγωνικού παλμού.

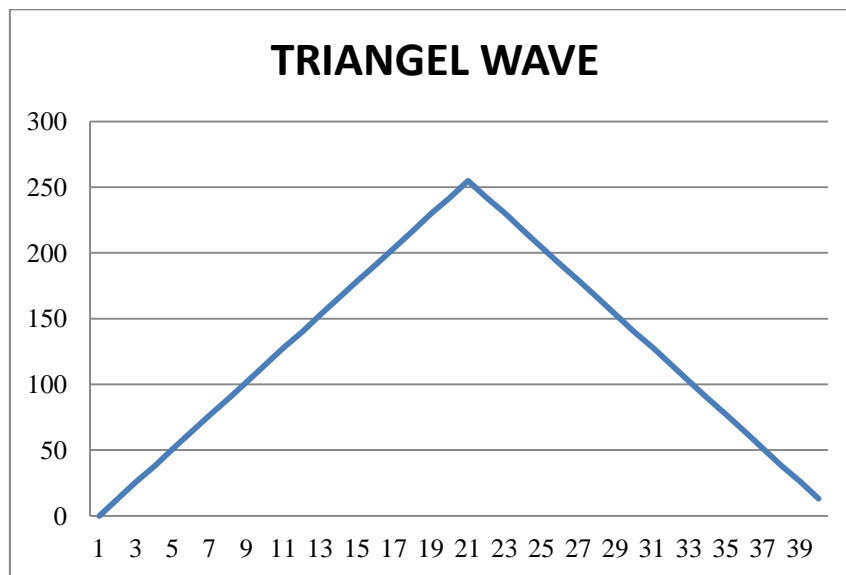
3.4.3 Τριγωνικό σήμα.

Στο τριγωνικό σήμα έχουμε:

Σχέση (1)/π για τα πρώτα 20 σημεία $y=ax+\beta$ όπου $a=1$ και $\beta=0$

και **Σχέση (1) /π+2** (διότι το σήμα δεν διέρχεται από την αρχή των αξόνων θα υποστεί μετατόπιση το σήμα $y=ax+\beta$ από την δεύτερη ημιπερίοδο. Όπου $a=-1$. Ευθεία με κλίση 1. $x=y$ και -1 όταν κατεβαίνει.

Και τέλος το αποτέλεσμα που θα βρούμε για κάθε τιμή το στρογγυλοποιούμε.



Εικόνα 3.14: Απεικόνιση τριγωνικού παλμού.

Πιο αναλυτικά η τιμές αναφέρονται στον παρακάτω πίνακα.

Τιμές	Φάση κάθε $2\pi/40$	$(\pi/20)/\pi$	$((\pi/20)/\pi)*255$	Σημεία
0	0	0	0	0
1	0,1570796	0,05	12,75	13
2	0,3141593	0,1	25,5	26
3	0,4712389	0,15	38,25	38
4	0,6283185	0,2	51	51
5	0,7853982	0,25	63,75	64
6	0,9424778	0,3	76,5	77
7	1,0995574	0,35	89,25	89
8	1,2566371	0,4	102	102
9	1,4137167	0,45	114,75	115
10	1,5707963	0,5	127,5	128
11	1,727876	0,55	140,25	140
12	1,8849556	0,6	153	153
13	2,0420352	0,65	165,75	166
14	2,1991149	0,7	178,5	179
15	2,3561945	0,75	191,25	191
16	2,5132741	0,8	204	204
17	2,6703538	0,85	216,75	217
18	2,8274334	0,9	229,5	230
19	2,984513	0,95	242,25	242
20	3,1415927	1	255	255
21	3,2986723	0,95	242,25	242
22	3,4557519	0,9	229,5	230
23	3,6128316	0,85	216,75	217
24	3,7699112	0,8	204	204
25	3,9269908	0,75	191,25	191
26	4,0840704	0,7	178,5	179
27	4,2411501	0,65	165,75	166
28	4,3982297	0,6	153	153
29	4,5553093	0,55	140,25	140
30	4,712389	0,5	127,5	128
31	4,8694686	0,45	114,75	115
32	5,0265482	0,4	102	102
33	5,1836279	0,35	89,25	89
34	5,3407075	0,3	76,5	77
35	5,4977871	0,25	63,75	64
36	5,6548668	0,2	51	51
37	5,8119464	0,15	38,25	38
38	5,969026	0,1	25,5	26
39	6,1261057	0,05	12,75	13

Πίνακας.(5): Τιμές τριγωνικού παλμού.

3.4.4 Sawtooth wave (Πριονωτή κυματομορφή).

Στην πριονωτή κυματομορφή χρησιμοποιώ την εκθετική συνάρτηση $y=A*[1-\exp(-x/\tau)]$ το π σε rad

A = είναι το μέγιστο δηλαδή 255 και τ είναι η σταθερά χρόνου.

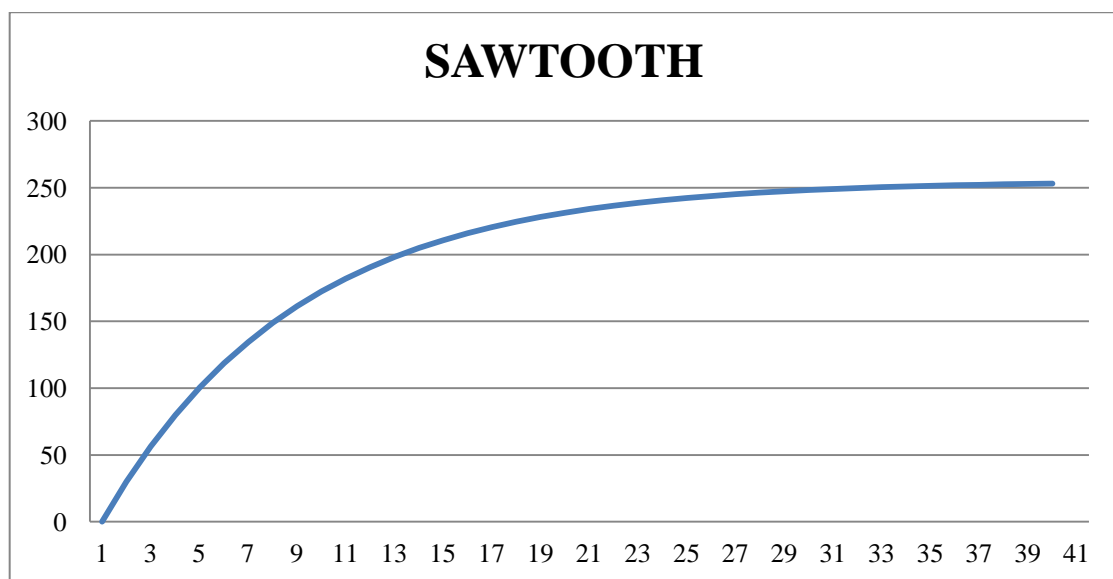
Περίπου 5τ διαρκεί ο παλμός. Οπότε το τ θα πρέπει να είναι [περίοδος/5].

Έτσι έχω

$$A=255, x= \text{Σχέση (1)}, \tau=2*\pi/5=6,28/5=1,256$$

$$\text{Άρα } y=255*[1-\exp(-\text{Σχέση (1)}/1,256)]$$

Και τέλος το αποτέλεσμα που θα βρούμε για κάθε τιμή το στρογγυλοποιούμε.



Εικόνα 3.15: Απεικόνιση sawtooth wave.

Πιο αναλυτικά η τιμές αναφέρονται στον παρακάτω πίνακα.

Τιμές	Φάση κάθε $2\pi/40$	$A*(1-EXP(-x/1,256))$	Σημεία
0	0	0	0
1	0,15708	29,97755712	30
2	0,314159	56,43098119	56
3	0,471239	79,77456591	80
4	0,628319	100,373901	100
5	0,785398	118,551598	119
6	0,942478	134,5923421	135
7	1,099557	148,7473517	149
8	1,256637	161,2383119	161
9	1,413717	172,2608466	172
10	1,570796	181,9875827	182
11	1,727876	190,5708529	191
12	1,884956	198,1450821	198
13	2,042035	204,8288921	205
14	2,199115	210,7269597	211
15	2,356194	215,9316562	216
16	2,513274	220,5244935	221
17	2,670354	224,5774012	225
18	2,827433	228,1538529	228
19	2,984513	231,3098604	231
20	3,141593	234,0948507	234
21	3,298672	236,5524401	237
22	3,455752	238,7211177	239
23	3,612832	240,6348476	241
24	3,769911	242,3236012	242
25	3,926991	243,8138266	244
26	4,08407	245,1288625	245
27	4,24115	246,289304	246
28	4,39823	247,3133251	247
29	4,555309	248,2169633	248
30	4,712389	249,0143706	249
31	4,869469	249,7180355	250
32	5,026548	250,3389782	250
33	5,183628	250,8869235	251
34	5,340708	251,3704529	251
35	5,497787	251,797139	252
36	5,654867	252,1736643	252
37	5,811946	252,5059256	253
38	5,969026	252,7991266	253
39	6,126106	253,0578592	253

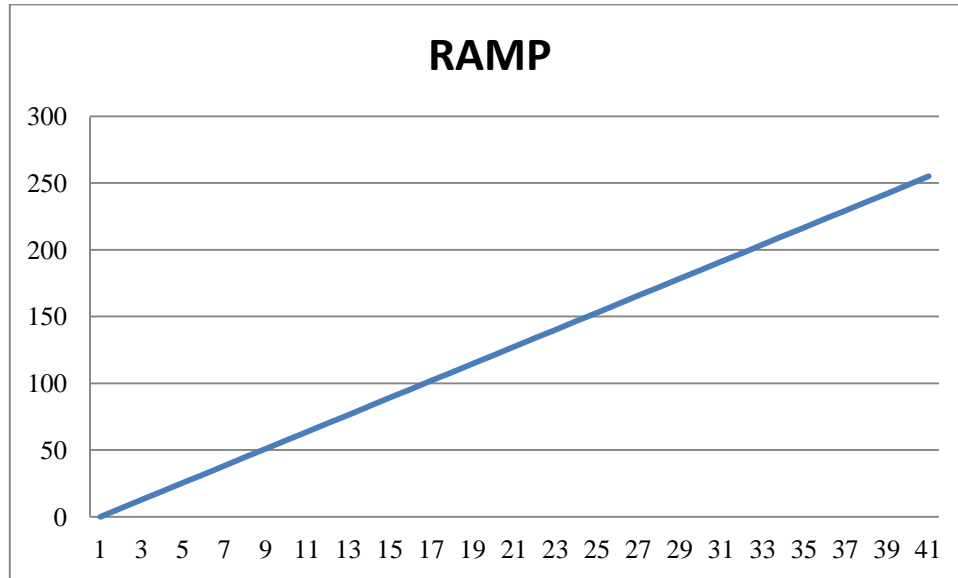
Πίνακας.(6): Τιμές sawtooth wave.

3.4.5 Κυματομορφή Ράμπας.

Στην κυματομορφή ράμπας έχω: Σχέση (1) /2π ←Σχέση (3)

255*Σχέση (3)

Και τέλος το αποτέλεσμα που θα βρούμε για κάθε τιμή το στρογγυλοποιούμε.



Εικόνα 3.16: Απεικόνιση κυματομορφής ράμπας.

Πιο αναλυτικά η τιμές αναφέρονται στον παρακάτω πίνακα.

Τιμές	Φάση κάθε 2π/40	$(\pi/20)/2\pi$	$((\pi/20)/2\pi)*255$	Σημεία
0	0	0	0	0
1	0,15708	0,025	6,375	6
2	0,314159	0,05	12,75	13
3	0,471239	0,075	19,125	19
4	0,628319	0,1	25,5	26
5	0,785398	0,125	31,875	32
6	0,942478	0,15	38,25	38
7	1,099557	0,175	44,625	45
8	1,256637	0,2	51	51
9	1,413717	0,225	57,375	57
10	1,570796	0,25	63,75	64
11	1,727876	0,275	70,125	70
12	1,884956	0,3	76,5	77
13	2,042035	0,325	82,875	83
14	2,199115	0,35	89,25	89
15	2,356194	0,375	95,625	96
16	2,513274	0,4	102	102
17	2,670354	0,425	108,375	108
18	2,827433	0,45	114,75	115
19	2,984513	0,475	121,125	121
20	3,141593	0,5	127,5	128
21	3,298672	0,525	133,875	134
22	3,455752	0,55	140,25	140
23	3,612832	0,575	146,625	147
24	3,769911	0,6	153	153
25	3,926991	0,625	159,375	159
26	4,08407	0,65	165,75	166
27	4,24115	0,675	172,125	172
28	4,39823	0,7	178,5	179
29	4,555309	0,725	184,875	185
30	4,712389	0,75	191,25	191
31	4,869469	0,775	197,625	198
32	5,026548	0,8	204	204
33	5,183628	0,825	210,375	210
34	5,340708	0,85	216,75	217
35	5,497787	0,875	223,125	223
36	5,654867	0,9	229,5	230
37	5,811946	0,925	235,875	236
38	5,969026	0,95	242,25	242
39	6,126106	0,975	248,625	249
40	6,283185	1	255	255

Πίνακας.(7): Τιμές κυματομορφής ράμπας.

3.5 Τεχνική Pulse Width Modulation (Διαμόρφωση Εύρους Παλμών).

Θα μπορούσε να αναρωτηθεί κάποιος το πώς μέσω ψηφιακών κυκλωμάτων παράγουμε ένα αναλογικό σήμα. Εδώ έρχεται η Pulse Width Modulation όπου θα το μεταφράζαμε καλύτερα ως Διαμόρφωση Εύρους Παλμών να μας βοηθήσει. Η διαμόρφωση εύρους παλμών είναι μια τεχνική που χρησιμοποιεί ένα ψηφιακό κύκλωμα για να δημιουργήσει ένα μεταβλητό αναλογικό σήμα. Το P.W.M. σαν έννοια είναι απλή, ανοίγει και να κλείνει ένα διακόπτη με ομοιόμορφα, επαναλαμβανόμενα χρονικά διαστήματα.

Σήμερα η τεχνική P.W.M. έχει μεγάλο εύρος εφαρμογών, όπως τον έλεγχο τάσης, ρεύματος, ηλεκτροκινητήρα, ισχύος, στα U.P.S, στους Inverters κ.α.

Με το να ελέγχουμε ψηφιακά τα αναλογικά κυκλώματα, το κόστος του συστήματος και η κατανάλωση ενέργειας μειώνονται δραστικά. Επιπλέον, πολλοί μικροελεγκτές και DSPs περιλαμβάνουν ήδη on-chip ελεγκτές P.W.M, καθιστώντας εύκολη την εφαρμογή.

Με λίγα λόγια, η τεχνική P.W.M είναι ένας πανίσχυρος τρόπος στο να ελέγχουμε το αναλογικό σήμα του κυκλώματος και τα συστήματά του, χρησιμοποιώντας την ψηφιακή έξοδο των μικροελεγκτών.

Ένα σήμα P.W.M χαρακτηρίζεται από τον κύκλο και χρόνο λειτουργίας του. Ο κύκλος λειτουργίας μετριέται σε επί τις εκατό (%) και η συχνότητα διαμόρφωσης σε Hz.

Η μέθοδος διαμόρφωσης εύρους παλμού (pulse width modulation – PWM) είναι μια πολύ διαδεδομένη μέθοδος ελέγχου σε συστήματα ισχύος. Η αρχή της μεθόδου βασίζεται στην διακοπτική λειτουργία των στοιχείων ισχύος. Το παραγόμενο σήμα που οδηγεί τα στοιχεία ελέγχου τα θέτει ή όχι σε αγωγιμότητα για συγκεκριμένο χρονικό διάστημα. Όσο μεγαλύτερος είναι ο χρόνος που το στοιχείο παραμένει σε αγωγιμότητα τόσο μεγαλύτερη είναι η μέση ισχύς που παρέχεται στο φορτίο. Συνήθως το παραγόμενο σήμα ελέγχου προέρχεται από κάποιο ψηφιακό επεξεργαστή ή κάποιο ολοκληρωμένο ψηφιακό κύκλωμα κατασκευασμένο για την συγκεκριμένη εφαρμογή. Το πλάτος του σήματος ελέγχου λαμβάνει 2 τιμές, μια για την κατάσταση αγωγιμότητας (ON) και μία για την κατάσταση διακοπής (OFF). Η απόλυτη τιμή του καθορίζεται από της ανάγκες της εφαρμογής.

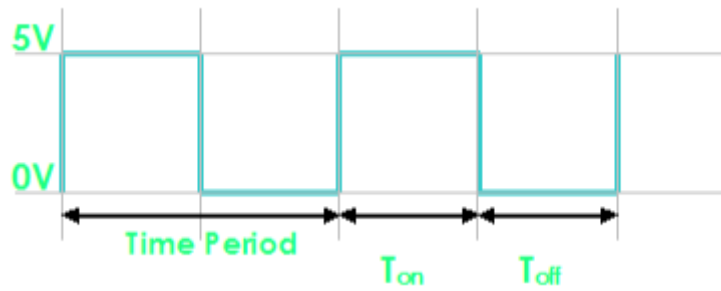
Η PWM λειτουργεί βάσει ενός ρολογιού (timer) ώστε να βγάζει τετραγωνικό παλμό κάποιας συχνότητας.

Με την εντολή analogWrite() αλλάζουμε το εύρος του παλμού στο χρόνο (duty cycle), η συχνότητα είναι σταθερή.

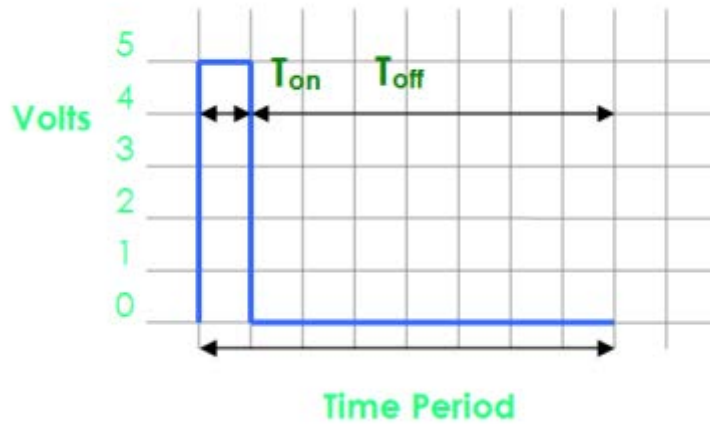
Το εύρος παλμού (duty-cycle) ορίζεται από τον τύπο:

$$duty\ cycle (\%) = \frac{T_{on}}{TimePeriod} * 100$$

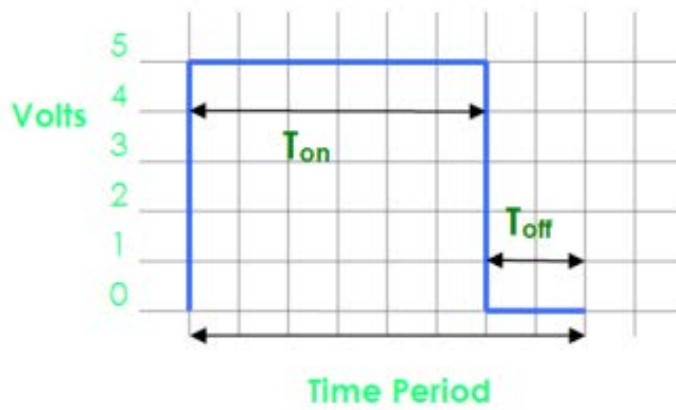
Παρακάτω ακολουθούν μερικά παραδείγματα διαμόρφωσης εύρους παλμού:



Εικόνα 3.17: Σήμα διαμορφωμένο κατά PWM με duty-cycle 50%.



Εικόνα 3.18: Σήμα διαμορφωμένο κατά PWM με duty-cycle 12.5%.



Εικόνα 3.19: Σήμα διαμορφωμένο κατά PWM με duty-cycle 75%.

Στην περίπτωση την δική μας Η συχνότητα για να αλλάξει πρέπει να επεμβούμε στον Prescaler του Timer. Ο καταχωρητής TCCR1B (για το pin9 και κάποια άλλα) στα 3 τελευταία bits αλλάζει τον prescaler.

```
TCCR1B = TCCR1B & 0b11111000 | 0x01;
```

το "TCCR1B & 0b11111000" κάνει τα 3 LSB Μηδέν. (πράξη AND)

Στο αποτέλεσμα κάνουμε OR (|) με την τιμή 0x01 (=0b00000001). Δηλαδή κάνουμε 1 το τελευταίο Bit.

Ίσως πιο κατανοητό είναι σε 2 βήματα,

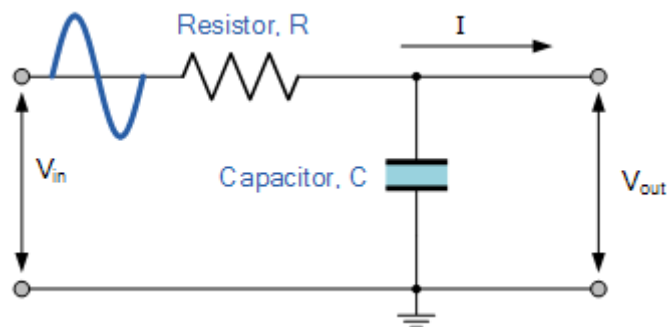
```
TCCR1B = TCCR1B & 0b11111000;  
TCCR1B = TCCR1B | 0b00000001;
```

Αυτό γίνεται περισσότερο κατανοητό στο επόμενο κεφάλαιο όπου εφαρμόζεται μέσα στον κώδικα του προγράμματος.

3.6 Υλοποίηση κατασκευής με κύκλωμα RC.

Πριν γίνει αναφορά στον κώδικα που θα προγραμματίσουμε τον μικροελεγκτή είναι αναγκαίο να υλοποιήσουμε το κύκλωμα με τα εξαρτήματα που αναφέραμε πιο πάνω.

Ένας πυκνωτής 100 nf με μία αντίσταση 1 kohm σε σειρά αποτελούν ένα RC κύκλωμα. Τα RC κυκλώματα χαρακτηρίζονται για την απόκρισή τους ως προς τη συχνότητα και ως προς το πλάτος.



Εικόνα 3.20: Κύκλωμα RC.

Τα RC κυκλώματα επιτρέπουν να περάσει το σήμα σε κάποιες συχνότητες, ενώ σε άλλες το κόβουν. Γι' αυτό, για την απόκρισή τους ως προς τη συχνότητα καλούνται φίλτρα. Διακρίνονται σε βαθυπερατά και υψιπερατά φίλτρα. Έτσι καθώς αυξάνει η συχνότητα, μειώνεται η χωρητική αντίσταση αφού είναι αντιστρόφως ανάλογη της συχνότητας.

$$f_c = \frac{1}{2\pi RC}$$

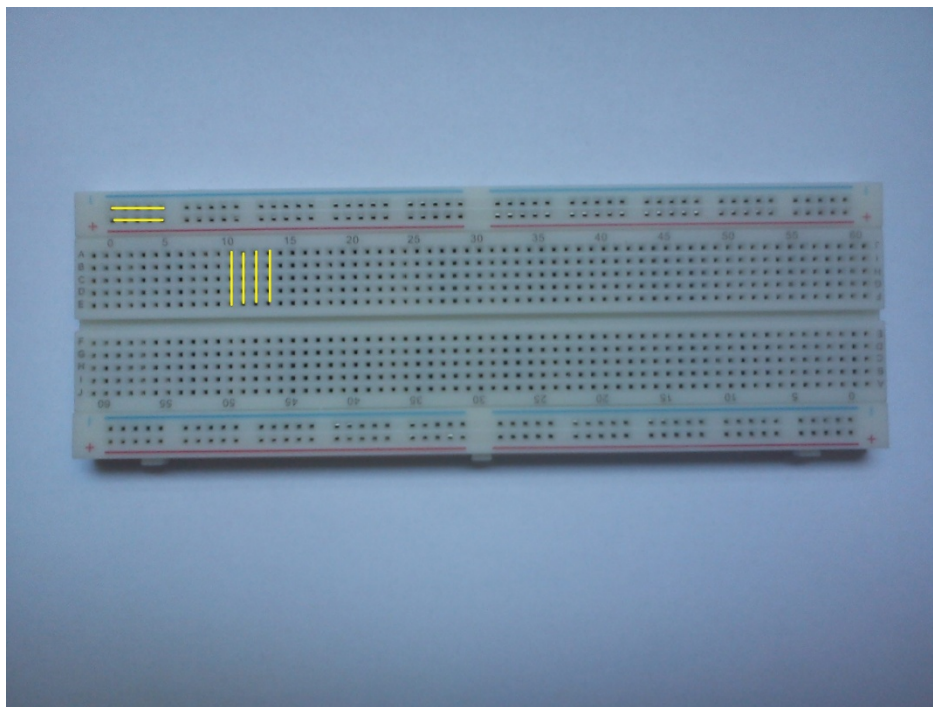
Εικόνα 3.21: Τύπος συχνότητας αποκοπής.

Για εμάς το f_c υπολογίζεται γύρο στα 1500 Hz.

Όταν αυξάνεται η χωρητική αντίσταση αυξάνεται και η συνολική αντίσταση του κυκλώματος και αφού η τάση είναι σταθερή, μικρότερη αντίσταση σημαίνει μεγαλύτερο ρεύμα κλάδου σειράς. Μεγαλύτερο ρεύμα σημαίνει μεγαλύτερη τάση στα άκρα της αντίστασης (αφού η τιμή της μένει σταθερή) και μικρότερη τάση στα άκρα του πυκνωτή (αφού η χωρητική αντίσταση μικραίνει με την αύξηση της συχνότητας) ώστε η συνολική τάση να είναι σταθερή και ίση με της πηγής.

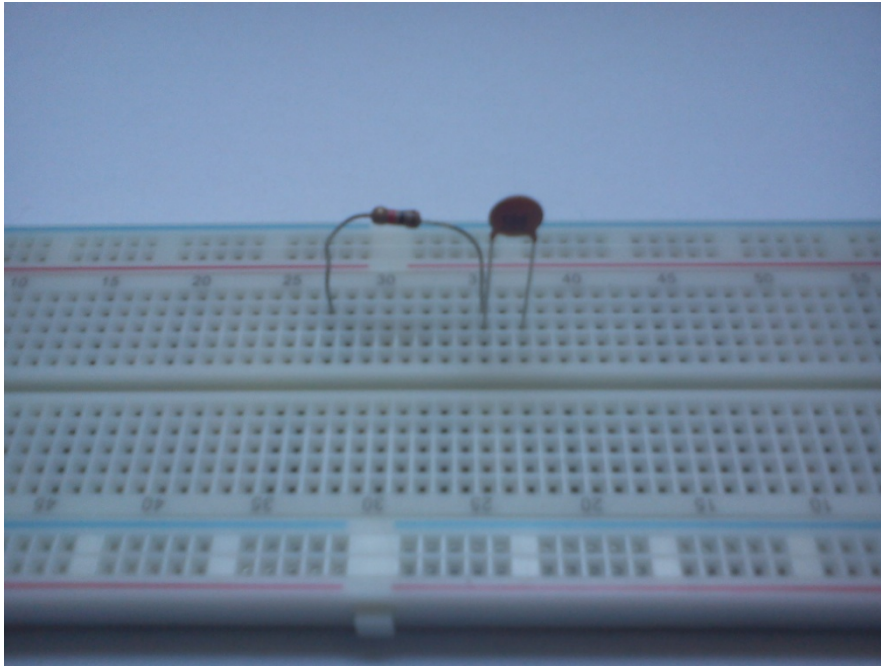
Προκειμένου να υλοποιήσουμε το κύκλωμα RC της εικόνας 3.20 θα χρησιμοποιήσουμε ένα breadboard όπως δίνεται στην εικόνα 3.22.

Στις εσωτερικές λωρίδες του breadboard οι θέσεις είναι βραχυκυκλωμένες μεταξύ τους κάθετα, ενώ στις εξωτερικές λωρίδες οι θέσεις είναι βραχυκυκλωμένες μεταξύ τους οριζόντια όπως δείχνουν και οι κίτρινες γραμμές στην εικόνα 3.22. Οι βραχυκυκλωμένες θέσεις αποτελούν κοινό σημείο (στις εσωτερικές λωρίδες ανά πέντε θέσεις κάθετα και στις εξωτερικές ανά είκοσι θέσεις οριζόντια).



Εικόνα 3.22: Breadboard.

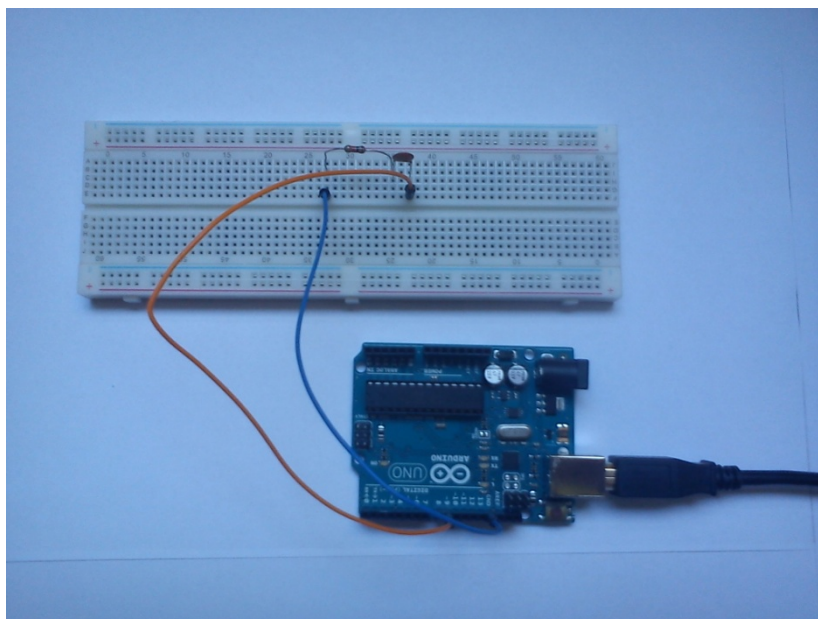
Για την πραγματοποίηση ενός πυκνωτή σε σειρά με μία αντίσταση αρκεί να τοποθετηθούν το ένα άκρο της αντίστασης σε βραχυκυκλωμένη θέση με το ένα άκρο του πυκνωτή όπως φαίνεται στην εικόνα 3.23.



Εικόνα 3.23: Πυκνωτής σε σειρά με μία αντίσταση.

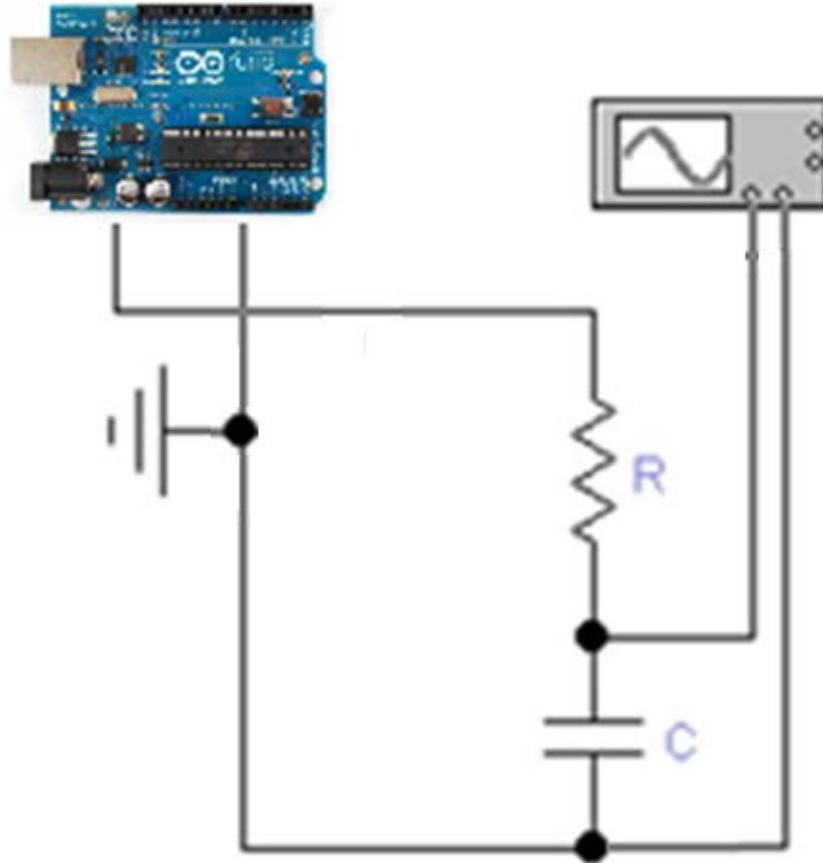
Επίσης μένει η συνδεσμολογία με τον μικροελεγκτή Arduino όπου και αυτός με το μέρος του στην συνέχεια θα ενωθεί με το ηλεκτρονικό μας υπολογιστή για να προγραμματιστή με την βοήθεια του προγράμματος Adriano IDE.

Από το pin 9 του μικροελεγκτή με ένα καλώδιο θα ενωθεί με το ελεύθερο σκέλος του πυκνωτή και το άλλο σκέλος της αντίστασης θα ενωθεί με το pin της γείωσης που διαθέτη ο μικροελεγκτής εικόνα 3.24.



Εικόνα 3.24: Σύνδεση κυκλώματος με Arduino.

Τέλος για την σωστή μέτρηση μέσω παλμογράφου αφού προγραμματίσουμε τον μικροελεγκτή θα είναι σύμφωνα με την εικόνα 3.25.



Εικόνα 3.24: Σωστή σύνδεση προκειμένου να οδηγηθεί η V_R προς μέτρηση στον παλμογράφο

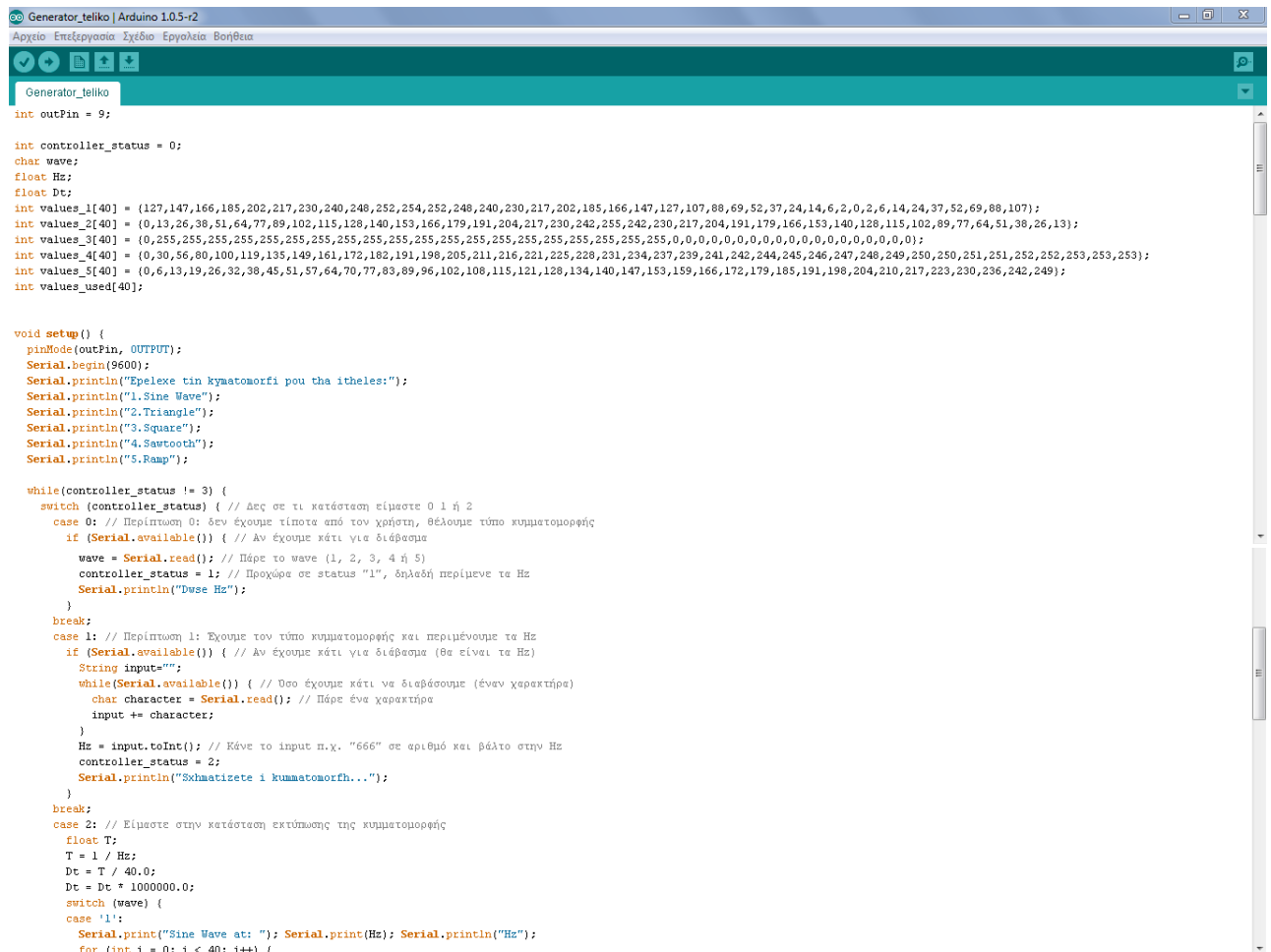
4. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

4.1 Λειτουργία γεννήτριας συχνοτήτων.

Στο πρόγραμμα αυτό θα αναπαρασταθούν οπτικά τα δεδομένα τα οποία αποστέλλονται από έναν χρήστη στον Arduino μικροελεγκτή.

Μία από τις πολλές δυνατότητες που μας δίνει το Arduino είναι η αμφίδρομη επικοινωνία μεταξύ υλικού και λογισμικού. Αυτό το πετυχαίνουν μέσω της σειριακής θύρας USB που συνδέεται με τον υπολογιστή και η οποία μας δίνει τη δυνατότητα αποστολής και λήψης δεδομένων σειριακά. Σε αυτή την εργασία θα δημιουργηθεί ένα monitoring tool, το οποίο θα μας δίνει τη δυνατότητα να επιλεγούμε την κυματομορφή που μας ενδιαφέρει και να δίνουμε μια τιμή σε Hz. Η δουλειά του Arduino θα είναι η αποστολή των δεδομένων από τον χρήστη, ενώ ο παλμογράφος θα είναι υπεύθυνος για τη αναπαράσταση-παρουσίαση αυτών των δεδομένων γραφικά και σε πραγματικό χρόνο.

4.2 Κώδικας προγράμματος και επεξήγηση.



```
Generator_teliko | Arduino 1.05-r2
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια

Generator_teliko
int outPin = 9;

int controller_status = 0;
char wave;
float Hz;
float Dt;
int values_1[40] = {127,147,166,185,202,217,230,240,248,252,254,252,248,240,230,217,202,185,166,147,127,107,88,69,52,37,24,14,6,2,0,2,6,14,24,37,52,69,88,107};
int values_2[40] = {0,13,26,38,51,64,77,89,102,115,128,140,153,166,179,191,204,217,230,242,255,242,230,217,204,191,179,166,153,140,128,115,102,89,77,64,51,38,26,13};
int values_3[40] = {0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int values_4[40] = {0,30,56,80,100,119,135,149,161,172,182,191,198,205,211,216,221,225,228,231,234,237,239,241,242,244,245,246,247,248,249,250,250,251,251,252,252,253,253};
int values_5[40] = {0,6,13,19,26,32,38,45,51,57,64,70,77,83,89,96,102,108,115,121,128,134,140,147,153,159,166,172,179,185,191,198,204,210,217,223,230,236,242,249};
int values_used[40];

void setup() {
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Epelexe tin kymatomorfi pou tha itheles:");
  Serial.println("1.Sine Wave");
  Serial.println("2.Triangle");
  Serial.println("3.Square");
  Serial.println("4.Sawtooth");
  Serial.println("5.Ramp");
}

while(controller_status != 3) {
  switch (controller_status) { // Δες σε τι κατάσταση είμαστε 0 1 ή 2
    case 0: // Περίπτωση 0: δεν έχουμε τίποτα από τον χρήστη, θέλουμε τύπο κυματομορφής
      if (Serial.available()) { // Αν έχουμε κάτι για διάβασμα
        wave = Serial.read(); // Πάρε το wave (1, 2, 3, 4 ή 5)
        controller_status = 1; // Προχώρα σε status "1", δηλαδή περίμενε τα Hz
        Serial.println("Dwse Hz");
      }
    }
    break;
  case 1: // Περίπτωση 1: Έχουμε τον τύπο κυματομορφής και περιμένουμε τα Hz
    if (Serial.available()) { // Αν έχουμε κάτι για διάβασμα (θε είναι τα Hz)
      String inpute="";
      while(Serial.available()) { // Όσο έχουμε κάτι να διαβάσουμε (έναν χαρακτήρα)
        char character = Serial.read(); // Πάρε ένα χαρακτήρα
        inpute += character;
      }
      Hz = inpute.toInt(); // Κάνε το input π.χ. "666" σε αριθμό και βάλτο στην Hz
      controller_status = 2;
      Serial.println("Sxmatizete i kymatomorfh...");
    }
    break;
  case 2: // Είμαστε στην κατάσταση εκτύπωσης της κυματομορφής
    float T;
    T = 1 / Hz;
    Dt = T / 40.0;
    Dt = Dt * 1000000.0;
    switch (wave) {
      case '1':
        Serial.print("Sine Wave at: "); Serial.print(Hz); Serial.println("Hz");
        for (int i = 0; i < 40; i++) {
```

```
        values_used[i] = values_1[i];
    }
    break;
case '2':
    Serial.print("Triangle at: "); Serial.print(Hz); Serial.println("Hz");
    for (int i = 0; i < 40; i++) {
        values_used[i] = values_2[i];
    }
    break;
case '3':
    Serial.print("Square at: "); Serial.print(Hz); Serial.println("Hz");
    for (int i = 0; i < 40; i++) {
        values_used[i] = values_3[i];
    }
    break;
case '4':
    Serial.print("Sawtooth at: "); Serial.print(Hz); Serial.println("Hz");
    for (int i = 0; i < 40; i++) {
        values_used[i] = values_4[i];
    }
    break;
case '5':
    Serial.print("Ramp at: "); Serial.print(Hz); Serial.println("Hz");
    for (int i = 0; i < 40; i++) {
        values_used[i] = values_5[i];
    }
    break;
default:
    Serial.println("Agnostos typos kumatomorfhs");
    controller_status = 0; // Σανέ στην αρχή να περιμένει τύπο κυματομορφής
    Serial.println("ksanadwse timh");
    break;
}
controller_status = 3; // Είμαστε στην κατάσταση εκτύπωσης τιμών απ' τον πίνακα
break;
}
delay(100);
}
TCCR1B = TCCR1B & 0b11111000 | 0x01;
}

void loop() {
    unsigned long T1=micros(); // τρέχουσα ώρα
    for (int i = 0; i < 40; i++) {

        analogWrite(outPin, values_used[i]); //εξάγεται το i-οστό δείγμα

        //-----Μέτρηση χρόνου-----
        while(1) // περιμένει συνεχώς μέχρι break
        {
            unsigned long T2=micros(); //τρέχουσα ώρα, το T1 είναι στην αρχή του βρόχου for
            // η διαφορά χρόνου είναι το T2-T1
            if((T2-T1)>=(i+1)*Dt) { break;}
            // όταν η διαφορά ισούται (ή ξεπεράσει) το (i+1)*Dt
            // (για το i-οστό δείγμα) σταματά το while
        }
        //-----
    }
}
}
```

Εικόνα 4.1: Κώδικας προγράμματος.

Παρακάτω ακολουθεί επεξήγηση του κώδικα σε συνδυασμό με τα σχόλια της εικόνας 4.1.

Για αρχή δηλώνουμε ότι θα χρησιμοποιήσουμε τη θήρα 9 του μικροελεγκτή με την εντολή `int outPin = 9;`. Με την εντολή `controller_status` απαριθμούμε την τιμή των καταστάσεων που βρισκόμαστε. Με αυτή την εντολή μπορούμε να διακρίνουμε καλύτερα τις περιπτώσεις που αναλύουμε μέσα στον κώδικα. Την εντολή `controller_status` τη δηλώνουμε ως ακέραιο αριθμό και παίρνει τιμές 0,1,2,3, δηλαδή 4 καταστάσεις μέσα στον κώδικα. Στη συνέχεια δηλώνουμε 3 μεταβλητές `wave`, `Hz`, `Dt` σε χαρακτήρα και πραγματικό αριθμό τις 2 τελευταίες αντίστοιχα. Η μεταβλητή `wave` θα πάρει μία τιμή από το χρήστη, η οποία αντιστοιχεί σε μία κυματομορφή εκ των πέντε που έχουμε αναφέρει. Η μεταβλητή `Hz` θα πάρει την τιμή της συχνότητας που θα δώσει ο χρήστης. Τέλος η μεταβλητή `Dt` αποδίδει το χρόνο μετά από υπολογισμούς, όπως είδαμε στο προηγούμενο κεφάλαιο, μεταξύ των σημείων σε κάθε κυματομορφή. Στο τελευταίο στάδιο των δηλώσεων γίνεται η δήλωση των πινάκων με τις τιμές που αντιστοιχούν σε κάθε κυματομορφή.

Μπαίνοντας στην `void setup()` ενεργοποιούμε τη θήρα 9 του μικροελεγκτή με την εντολή `pinMode(outPin, OUTPUT)`; Στη συνέχεια στον κώδικα μας κάνουμε χρήση της συνάρτησης `Serial.begin()` και `Serial.println()`. Τις συναρτήσεις αυτές τις χρησιμοποιούμε για να στείλουμε σειριακά κάποιο μήνυμα, στην περίπτωση μας αυτό το μήνυμα εμφανίζεται στην σειριακή οθόνη του Arduino IDE. Με την εντολή `Serial.begin()` δηλώνουμε στον μικροελεγκτή με τι ρυθμό θα στέλνει τα δεδομένα στην σειριακή θύρα, εμείς θα χρησιμοποιούμε πάντα ρυθμό μετάδοσης 9600bps: `Serial.begin(9600)`; Με τη συνάρτηση `Serial.println()` τυπώνουμε σε μήνυμα στη σειριακή οθόνη τα είδη των κυματομορφών που θα επιλέξει ο χρήστης με τη βοήθεια της μεταβλητής `wave`, όπως φαίνεται στην εικόνα 4.1. Στη συνέχεια προγραμματίζουμε τη διαδικασία αντιστοιχίας μεταξύ τιμών και μεταβλητών όπως και πράξεων, όπου σε κάθε βήμα αναφέρω το κάθε σχόλιο στην αντίστοιχη εντολή, εικόνα 4.1.

Στο παρακάτω κομμάτι κώδικα, γίνεται η εκτύπωση σε μήνυμα στη σειριακή οθόνη ο τύπος της κυματομορφής, η συχνότητα που έχει δώσει ο χρήστης και η αντιστοιχία των τιμών του πίνακα με τη βοήθεια της επαναληπτικής δομής `for`. Στο συγκεκριμένο κομμάτι βλέπουμε το μήνυμα για το ημίτονο. Αντίστοιχα έτσι γίνεται και για τα υπόλοιπα.

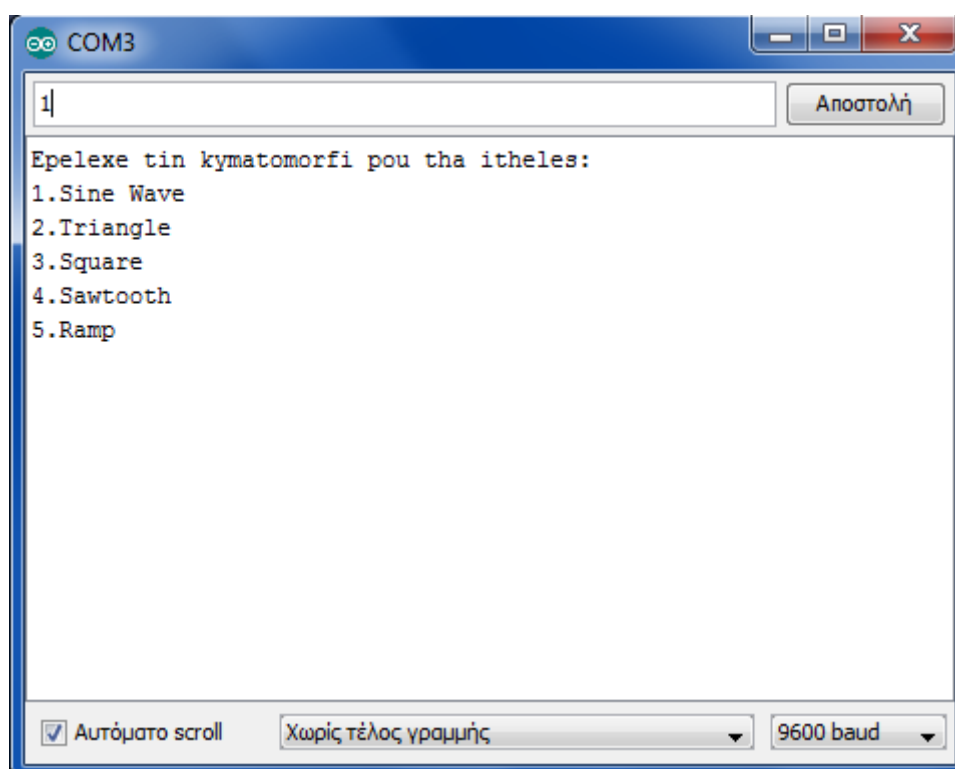
```
case '1':
    Serial.print("Sine Wave at: "); Serial.print(Hz); Serial.println("Hz");
    for (int i = 0; i < 40; i++) {
        values_used[i] = values_1[i];
    }
    break;
```

Τέλος μέσα στην `void loop()` έχουμε την έξοδο των τιμών, κάθε πίνακα που αντιστοιχεί σε κυματομορφή, στη θήρα του μικροελεγκτή με τη βοήθεια της συνάρτησης `analogWrite(outPin, values_used[i])`; Επίσης κάνουμε μία μέτρηση χρόνου με τις εντολές που βλέπουμε στο τέλος του κώδικα της εικόνας 4.1 για να μπορέσουμε να έχουμε όσο το δυνατόν λιγότερη διαφορά στη συχνότητα αυτή που δείχνει ο παλμογράφος με αυτή που δίνει ο χρήστης μέσω της σειριακής οθόνης.

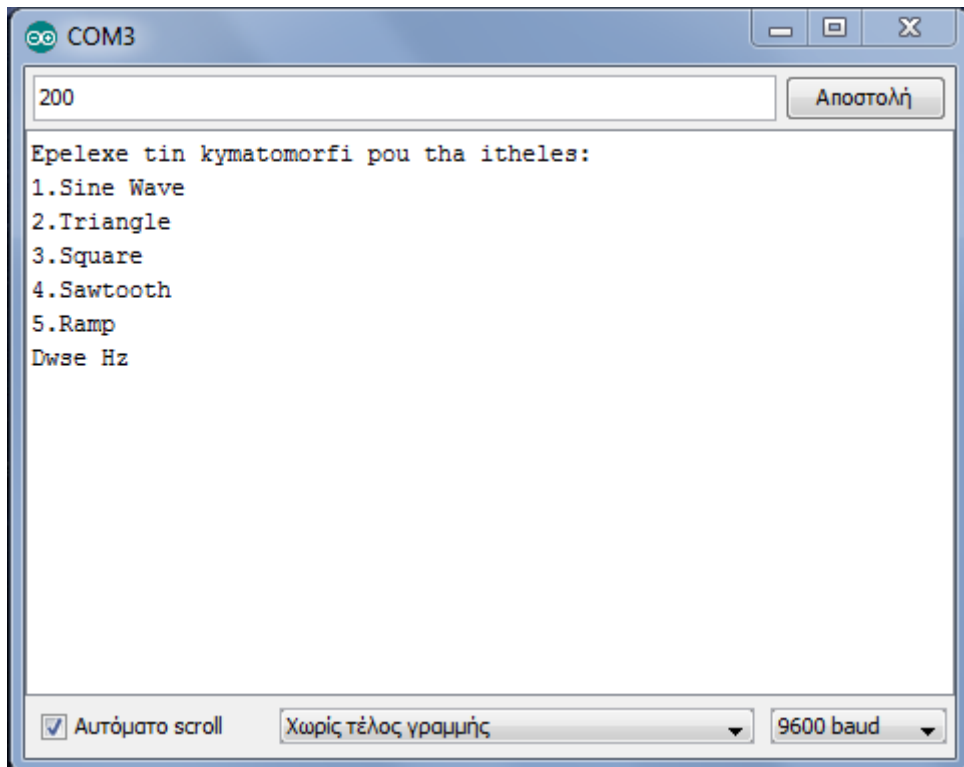
5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.

Η εργασία είχε ως στόχο την δημιουργία μιας γεννήτριας συναρτήσεων η οποία όπως είδαμε κατασκευαστική από απλά υλικά και κύριος οικονομικά. Θα μπορούσαμε να πούμε επίσης ότι η κατασκευή αυτή θα μπορούσε να αποτελέσει μια εύκολη λύση για οιονδήποτε θα ήθελε να έχει σε πολύ μικρό χρονικό διάστημα μια γεννήτρια συναρτήσεων και να την χρησιμοποιήσει με μεγάλη αξιοπιστία αφού οι εικόνες οι οποίες ακλουθούν θα δούμε το τελικό αποτέλεσμα το οποίο είναι αυτό που περιμέναμε δηλαδή την εκτέλεση του κώδικα, η επιλογή κυματομορφής μέσω σειριακής οθόνης και η απεικόνιση στον παλμογράφο.

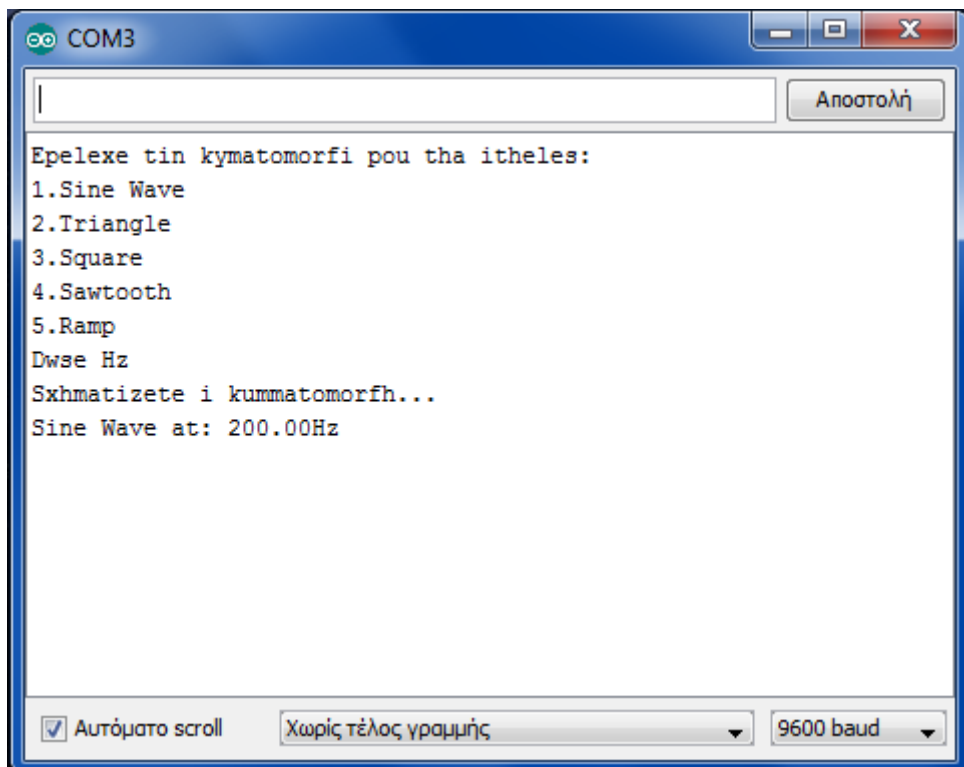
Για αρχή απαραίτητη προϋπόθεση είναι να έχουν εφαρμοστή όλα όσα αναφέραμε παραπάνω σχετικά με την υλοποίηση της κατασκευής. Σε δεύτερη φάση ανοίγουμε το λειτουργικό του μικροελεγκτή και εκτελούμε τον κώδικα. Αφού εκτελεστή ο κώδικας ανοίγουμε την σειριακή οθόνη και επιλεγούμε μια κυματομορφή από τις πέντε στην συχνότητα που θέλουμε. Όπως φαίνετε στις εικόνες παρακάτω για αρχή θα απεικονίσουμε την κυματομορφή ημιτόνου στην συχνότητες των 200 Hz και μετά ακολουθούν οι υπόλοιπες κυματομορφές στην ίδια συχνότητα όπως έγινε η απεικόνιση στον παλμογράφο.



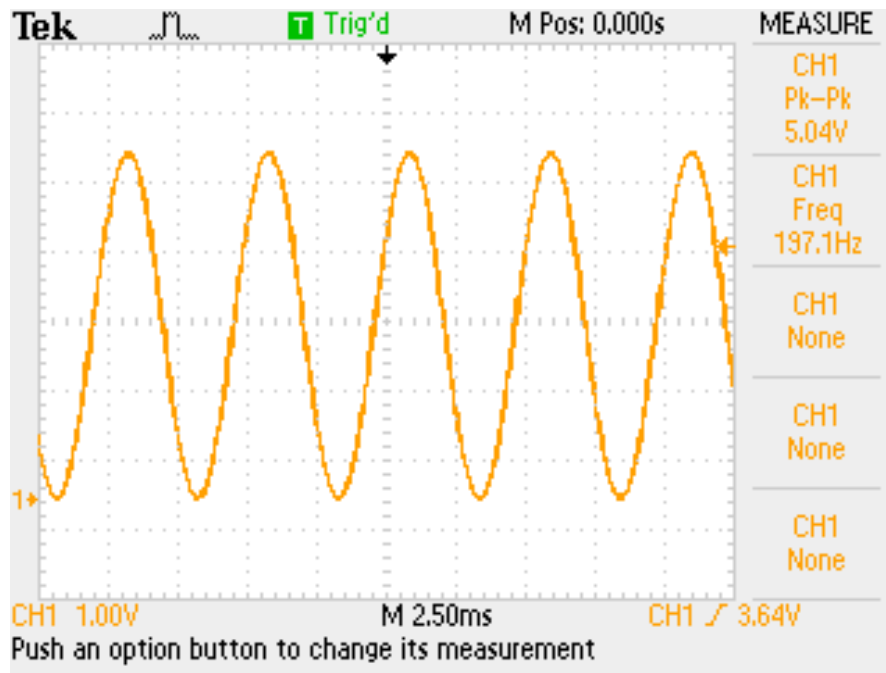
Εικόνα 5.1: Επιλογή κυματομορφής.



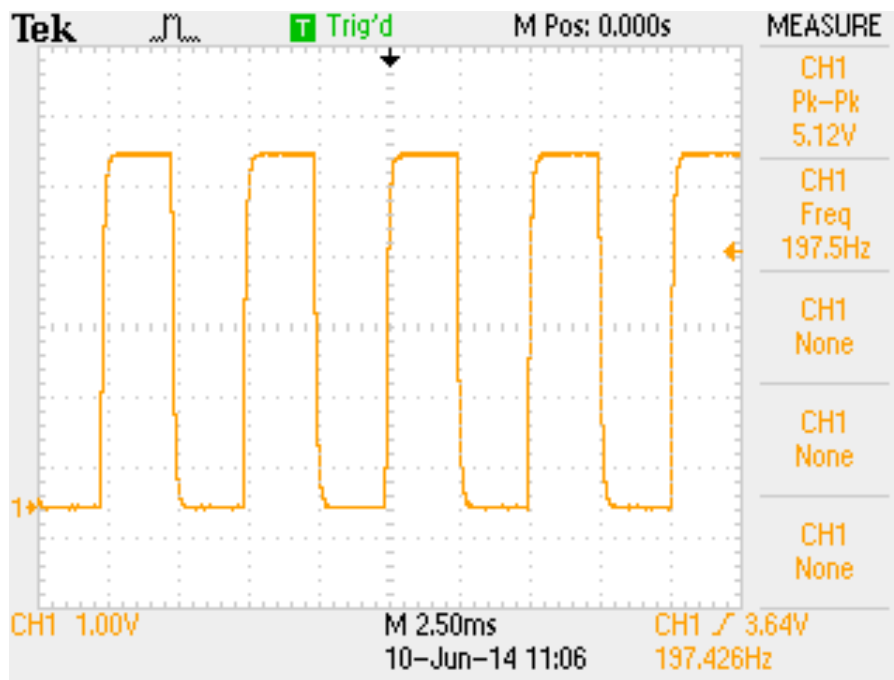
Εικόνα 5.2: Επιλογή συχνότητας.



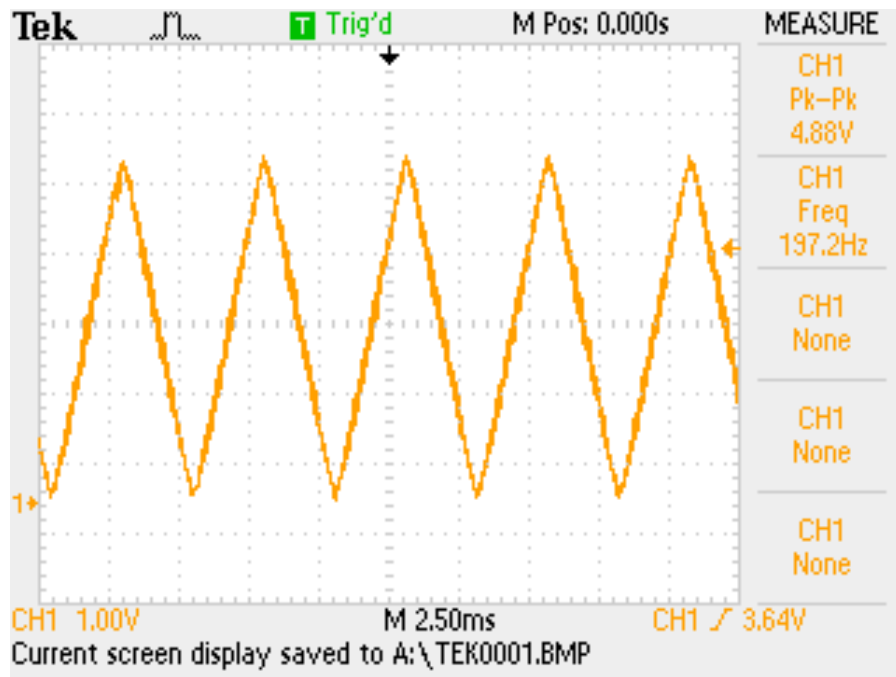
Εικόνα 5.3: Σχηματισμός κυματομορφής.



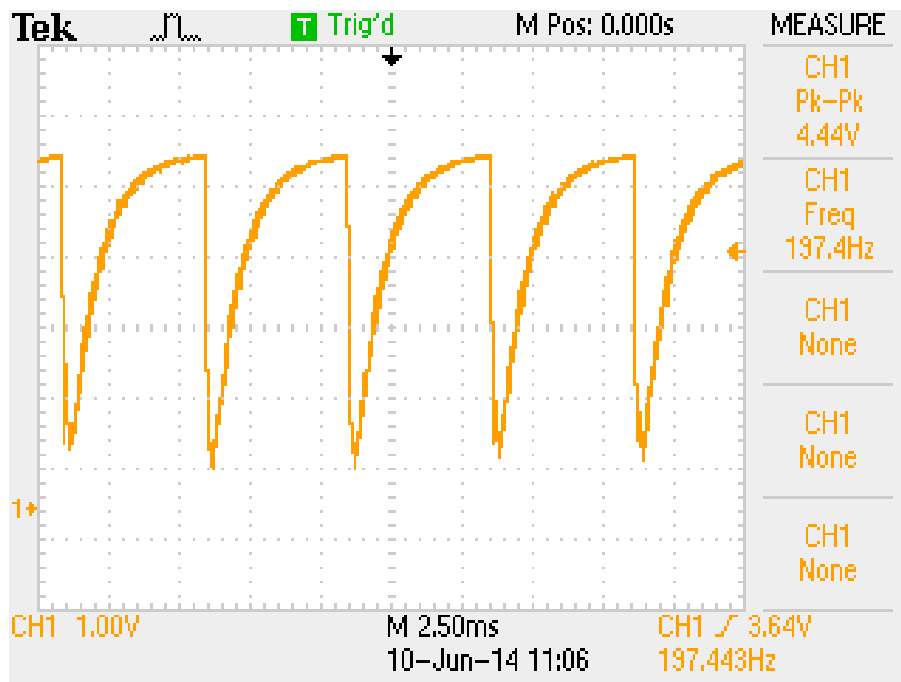
Εικόνα 5.4: Απεικόνιση ημιτόνου στον παλμογράφο



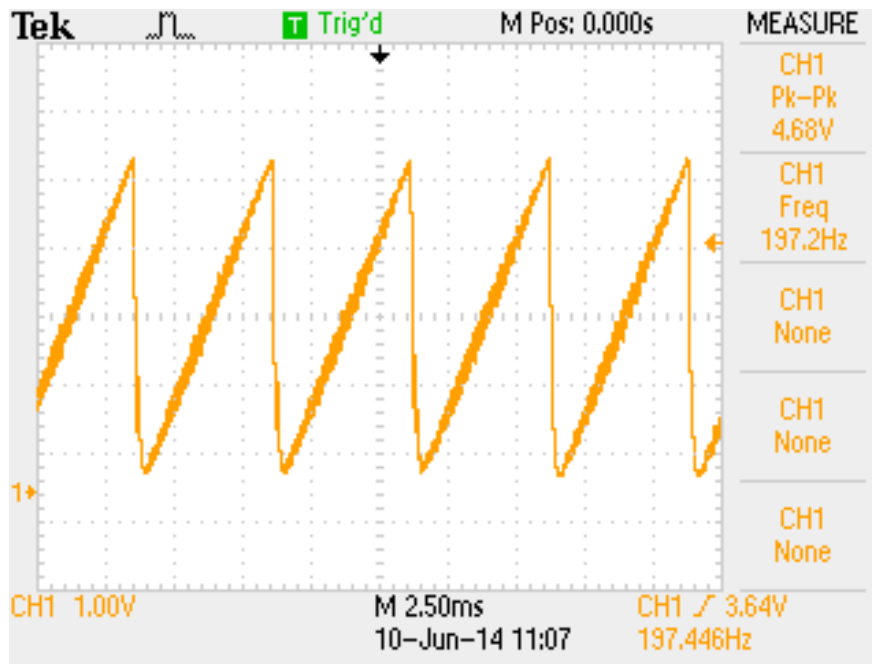
Εικόνα 5.5: Απεικόνιση τετραγωνικού παλμού στον παλμογράφο



Εικόνα 5.6: Απεικόνιση τριγωνικού παλμού στον παλμογράφο.



Εικόνα 5.7: Απεικόνιση saw tooth wave στον παλμογράφο.



Εικόνα 5.8: Απεικόνιση σήματος ράμπας στον παλμογράφο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- <http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>
- Πτυχιακή εργασία «Σχεδίαση και υλοποίηση γεννήτριας συναρτήσεων με μικροελεγκτή» Τεχνολογικό Εκπαιδευτικό Ίδρυμα Σερρών, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών Γκλαβινη Τατιανή, Σέρρες 2009
- Τεχνολογία Γ΄ Λυκείου Ηλεκτρονική Μνήμη και Μικροελεγκτές
- Πτυχιακή εργασία «Η διδασκαλία εκπαιδευτικής ρομποτικής με τη χρήση μικροελεγκτών (πχ Arduino, Pic)» Ίδρυμα Λάρισας Τεχνολογικών Εφαρμογών, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών.. Βασιλική Παυλή Λάρισα 2013
- <http://el.wingwit.com/Hardware/computer-drives-storage/47393.html>(Εισαγωγή στα συστήματα μικροελεγκτών)
- http://nereus.mech.ntua.gr/Documents/pdf_ps/pm4.pdf
- Εργασία Νικολάου Γιαννακόπουλου Προγραμματιζόμενοι ελεγκτές και εφαρμογή αλγορίθμων σε παραδείγματα ρομποτικής μέσα από συγκεκριμένες δραστηριότητες.
- Διπλωματική Εργασία - Μαρτίνης Στέλιος «Οικιακός Αυτοματισμός με χρήση Μικροελεγκτή» Πανεπιστήμιο Αιγαίου Σχολή Θετικών Επιστημών Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων Σάμος 2013
- Εργασία Νικολάου Νικολαΐδη Εργαστήριο Μικροϋπολογιστών ,Τμήμα Αυτοματισμού, Αλεξανδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης
- <http://arduino.cc/en/Tutorial/HomePage>
- <http://www.me.umn.edu/courses/me2011/arduino/arduinoGuide.pdf>
- <http://deltahacker.gr/2009/08/01/arduino-intro/>
- Διπλωματική εργασία Δημοσθένη Μπότσαρη Συσκευή USB για μέτρηση μαγνητικού πεδίου με αισθητήρα Hall(gaussmeter Hall) Θεσσαλονίκη 2013
- http://arch.icte.uowm.gr/mdasyg/pdfs/dimitriou_thesis.pdf
- <http://el.wikipedia.org/wiki/Arduino>
- http://www.teicm.gr/repository/repository/syggkentrwsh_ylikou/PTYXIAKES_ERGASIES/PTYXIAKES_ERGASIES_PLHROFORIKHS_2012/PTYX_KATHIGITWN_TOMEA_3/Vologiannidis/Fellopoulos_Spyrou/Fellopoulos_Spyrou.pdf
- <http://th.wikispaces.com/file/view/%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%83%CF%84%CE%BFArduino.pdf/432670936/%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%83%CF%84%CE%BFArduino.pdf>
- http://arch.icte.uowm.gr/mdasyg/pdfs/dimitriou_thesis.pdf
- <http://www.microplanet.gr/tutorials/microcontrollers/arduino>
- <http://www.scribd.com/doc/117118381/Arduino>
- <http://vivliothmyy.ee.auth.gr/2179/1/gaussmeter.pdf>
- http://chaos.c4lab.el.teithe.gr/kioskeridis/Electrical_Measurements_05_Oscilloscopes.pdf
- <http://ekfe.eyr.sch.gr/physics/palmografos.pdf>
- <http://el.wikipedia.org/wiki/%CE%A0%CF%85%CE%BA%CE%BD%CF%89%CF%84%CE%AE%CF%82>

- <http://ebooks.edu.gr/modules/ebook/show.php/DSGL-B134/513/3336,13484/>
- <http://coolweb.gr/antistasi-kiklomata-pos-leitourgei/>
- <https://robotstore.gr/el/>
- <https://sites.google.com/site/electriccurrentonhuman/home/epitoseis/factors/e-tase-tou-elektrikou-reumatos>
- http://users.ntua.gr/apolyk/Exam_Notes.pdf
- <http://www.sz7xan.gr/radiotheory.htm#AAC>
- <file:///C:/Users/Spytzi/Downloads/P.W.M%20K%E2%80%99%20SWITCHING%20%CE%A4%CE%95%CE%A7%CE%9D%CE%9F%CE%9B%CE%9F%CE%93%CE%99%CE%95%CE%A3.pdf>
- https://okeanis.lib.teipir.gr/xmlui/bitstream/handle/123456789/1136/aut_00597.pdf?sequence=1
- <http://arduino-info.wikispaces.com/Arduino-PWM-Frequency>
- http://users.ntua.gr/apolyk/RC_Notes.pdf
- http://www.electronics-tutorials.ws/filter/filter_2.html