ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

Μελέτη και υλοποίηση γεννήτριας σήματος, βασισμένη στην πλατφόρμα Arduino

Πτυχιακή Εργασία

Μεϊμαρίδης Δημήτριος
Πάλλης Δημήτριος

Επιβλέπων :Ιωάννης Α. Μαδεμλής Ηλεκτρονικός Μηχανικός

ΣΕΡΡΕΣ, ΜΑΙΟΣ 2013

Περιεχόμενα

Περίληψη:

Στα πλαίσια αυτής της πτυχιακής εργασίας γίνεται η σχεδίαση και υλοποίηση μιας γεννήτριας συναρτήσεων που συνδέεται με το προγραμματιζόμενο κύκλωμα Arduino, η οποία παράγει σήματα όπως τετραγωνικό παλμό, τριγωνικό, πριονωτό σήμα και σήμα ημιτόνου. Επίσης δίνεται δυνατότητα ρύθμισης της συχνότητας του πλάτους και του τρόπου λειτουργίας από τον χρήστη.

Για την υλοποίηση του κυκλώματος χρησιμοποιήθηκε το Arduino Uno, το οποίο διαθέτει κατάλληλα περιφερειακά, όπως Raster υλοποίησης κυκλώματος μονάδες εισόδου – εξόδου αναλογικού και ψηφιακού σήματος, ποτενσιόμετρα, αντιστάσεις, leds, και παλμογράφος για την απεικόνιση των σημάτων στο πεδίο του χρόνου και της συχνότητας.

Η σύνδεση του Arduino προωθείται μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Χρησιμοποιούμε το λογισμικό Arduino Software για την αποστολή δεδομένων με σκοπό τη ρύθμιση των συχνοτήτων και του τρόπου λειτουργίας από τον χρήστη.

ΠΡΟΛΟΓΟΣ

Ο σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη και υλοποίηση μιας γεννήτριας συναρτήσεων με την πλατφόρμα Arduino, η οποία παράγει τετραγωνικό παλμό, τριγωνικό παλμό, ημιτονικό σήμα και πριονωτή κυματομορφή.

Στο 1^{ο} κεφάλαιο, μελετάται η βασική αρχιτεκτονική και τον προγραμματισμό του Arduino, που χρησιμοποιήθηκε στην παρούσα εφαρμογή για την υλοποίηση της γεννήτριας.

Στο 2^{ο} κεφάλαιο αναφέρονται τα 4 σήματα τα οποία θα παράγει η γεννήτρια.

Στο 3^{ο} κεφάλαιο μελετάται ο μετατροπέας R-2R. Το κύκλωμα αυτό είναι μετατροπέας ψηφιακού σήματος σε αναλογικό και μετατρέπει τα δεδομένα που είναι σε ψηφιακή μορφή και εξάγονται στα pins 0, 1, 2, 3, 4, 5, 6, 7, σε αναλογική τάση για την παραγωγή παλμού ημιτόνου και πριονωτού παλμού.

Στο 4^{ο} κεφάλαιο παρουσιάζεται το κύκλωμα της εφαρμογής μας, και αναλύεται ο κώδικας της εφαρμογής, που γράφηκε στη γλώσσα Wiring, μια παραλλαγή C/C++. Επίσης, παρουσιάζεται σύντομα το λογισμικό Arduino Software, στο οποίο γράψαμε μια εφαρμογή host, ώστε να υπάρχει δυνατότητα επιλογής τρόπου λειτουργίας (τετραγωνικός παλμός, σήμα ημιτόνου ή πριονωτό σήμα) και ρύθμισης της συχνότητας των κυματομορφών.

Το 5^{ο} κεφάλαιο είναι αφιερωμένο στις μετρήσεις που καταγράψαμε και στις παρατηρήσεις μας, καθώς και στα συμπεράσματα.

Η εργασία εκπονήθηκε στα εργαστήρια του τομέα Αρχιτεκτονικής Υπολογιστών και Βιομηχανικών Εφαρμογών του Τμήματος Πληροφορικής και Επικοινωνιών του ΤΕΙ Σερρών.

ΚΕΦΑΛΑΙΟ 1ο

1.1 Στόχος

Σκοπός της συγκεκριμένης εργασίας είναι η επίδειξη του προγραμματισμού της πλατφόρμας Arduino και η κατασκευή κυκλώματος για τη δημιουργία μιας γεννήτριας συναρτήσεων η οποία θα παράγει τα σήμα ημιτονικό, τριγωνικό, τετραγωνικό και πριονωτό. Χρειάστηκε να μελετηθεί η αρχιτεκτονική και το προγραμματιστικό μέρος της πλατφόρμας Arduino για την κατασκευή και λειτουργία της γεννήτριας. Επίσης μελετήθηκαν και οι ιδιότητες και ιδιαιτερότητες του κάθε σήματος.

1.2 Ιστορικά Στοιχεία

Το Arduino δημιουργήθηκε το 2005, από μαθητές με πολύ λίγα έξοδα σε σχέση με άλλα παρόμοια συστήματα που υπήρχαν εκείνη την εποχή. Ήταν τόσο εύχρηστο που μέχρι τον Μαίο του 2011 υπήρχαν πάνω από 300.000 Arduino. Arduino σημαίνει «γενναίος φίλος» στα Ιταλικά, το όνομα αυτό δόθηκε από τους εφευρέτες του έργου Massimo Banzi και David cuartieless.
Το Arduino, είναι βασισμένο σε μία πλατφόρμα ανοικτού κώδικα που φέρει το όνομα wiring platform

1.3 Γενικά Στοιχεία

Ο Arduino θα λέγαμε ότι είναι ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό Ηλεκτρονικό Υπολογιστή. Είναι ανοιχτού υλικού και λογισμικού και βασίζεται σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να τον προγραμματίσουμε μέσα από ένα απλό περιβάλλον ανάπτυξης. Το Arduino,μπορεί να χρησιμοποιηθεί για να αναπτύξουμε διαδραστικά αντικείμενα, να δεχτούμε εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχουμε διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου. Οι πλακέτες μπορούν εύκολα να συναρμολογηθούν ακόμη και από έναν αρχάριο ή να αγοραστούν μονταρισμένες. Το περιβάλλον ανάπτυξης του λογισμικού βασίζεται στην γλώσσα προγραμματισμού Processing και την γλώσσα προγραμματισμού Wiring, οι οποίες είναι ανοιχτού κώδικα (open source).

1.4 Αρχιτεκτονική

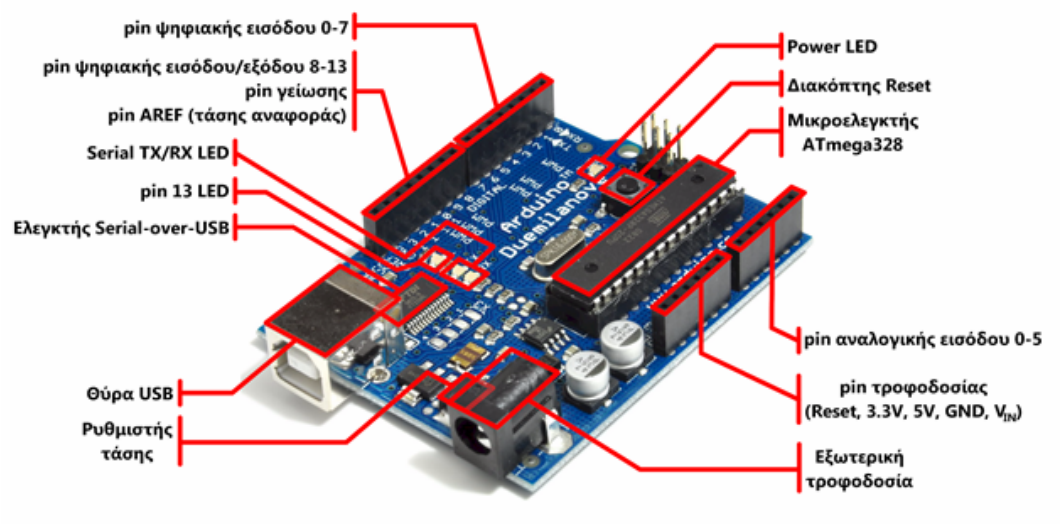
Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, ο οποίος χρονίζεται στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων: 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

Περιέχει 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset.

Επίσης περιέχει 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή . Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σε μεταγλωττισμένη μορφή).

1.4.1 Είσοδοι – Έξοδοι

Το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορούμε λόγου χάρη να ανάψουμε και να σβήσουμε ένα LED που έχουμε συνδέσει στο συγκεκριμένο pin.

Αν πάλι ρυθμίσουμε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά, μπορούμε με την κατάλληλη εντολή να διαβάσουμε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχουμε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγου χάρη είναι δυνατό το διάβασμα της κατάστασης ενός διακόπτη). Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα: Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά μας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγου χάρη το πρόγραμμά μας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά μας ενεργοποιήσουμε το σειριακό interface, χάνουμε 2 ψηφιακές εισόδους/εξόδους.

Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούμε να τα ρυθμίσουμε μέσα από το πρόγραμμά μας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι 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.

Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορούμε να συνδέσουμε λόγου χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξουμε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχουμε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπές ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβουμε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρούμε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή.

Για παράδειγμα, μπορούμε να τροφοδοτήσουμε ένα από αυτά με μια τάση την οποία μπορούμε να αλλάζουμε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref η οποία, αν δεν κάνουμε κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά μας μπορούμε να «διαβάσουμε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσουμε το pin AREF με 3.3V και στην συνέχεια δοκιμάσουμε να διαβάσουμε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα μας επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

1.5 Προγραμματισμός

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 την C, στην γλώσσα του Arduino μπορούν να χρησιμοποιηθούν

ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, οι ίδιοι τύποι δεδομένων και οι ίδιοι 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.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες ώστε να έχει την γενική δομή:

```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

void setup()
{
// ...
}

void loop()
{
// ...
}

// Υπόλοιπες συναρτήσεις...
```

Η βασική ρουτίνα setup() εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα loop() περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος while (true).

Ας ρίξουμε μια ματιά στην δημιουργία και την σύνταξη ενός νέου sketch. Αυτό αρχίζει με την έκφραση #include στην κορυφή του sketch στην οποία ενσωματώνονται οι βιβλιοθήκες μας. Στην συνέχεια ακλουθούν οι global μεταβλητές και αρχικοποίηση αντικειμένων τα οποία συναντώνται στην ρουτίνα setup( ). Η συνάρτηση setup( ) χρησιμοποιείται για την αναφορά στα pins του Arduino μέσο των αρχικοποιημένων αντικειμένων παραπάνω. Έτσι επιτυγχάνεται μια φυσική σύνδεση αυτών των δυο. Ένα παράδειγμα θα μπορούσε να είναι int board _led=13 πριν την συνάρτηση setup( ). Η εντολή ενεργοποιεί για χρήση το pin 13.Στην συνέχεια μέσα στην setup( ) θα το χρησιμοποιούσαμε ως έξοδο για παράδειγμα με την εντολή pinMode (on board_led,OUTPUT). Μετά την αρχικοποίηση των μεταβλητών και την συνάρτηση setup ακολουθεί το κύριο κομμάτι με την συνάρτηση loop( ).

Στο σημείο αυτό το sketch αναμένω να γίνει ένα γεγονός για να προχωρήσει ή να επαναλάβει μια συγκεκριμένη εργασία.

1.5.1 Γλώσσα Προγραμματισμού

Η πλακέτα Arduino Uno μπορεί να προγραμματιστεί με το λογισμικό Arduino. Οι μικροεπεξεργαστές ATmega328 στο Arduino παρέχονται με bootloader ο οποίος επιτρέπει την μεταφόρτωση νέου κώδικα σε αυτό χωρίς τη χρήση ενός εξωτερικού προγραμματιστή υλικού. Επικοινωνεί με τη χρήση του αρχικού STK500 πρωτοκόλλου. Μπορεί επίσης να παρακαμφθεί και το bootloader και να προγραμματιστεί ο μικροεπεξεργαστής μέσω της επιγραφής ICSP (προγραμματισμός σε σειριακό κύκλωμα).

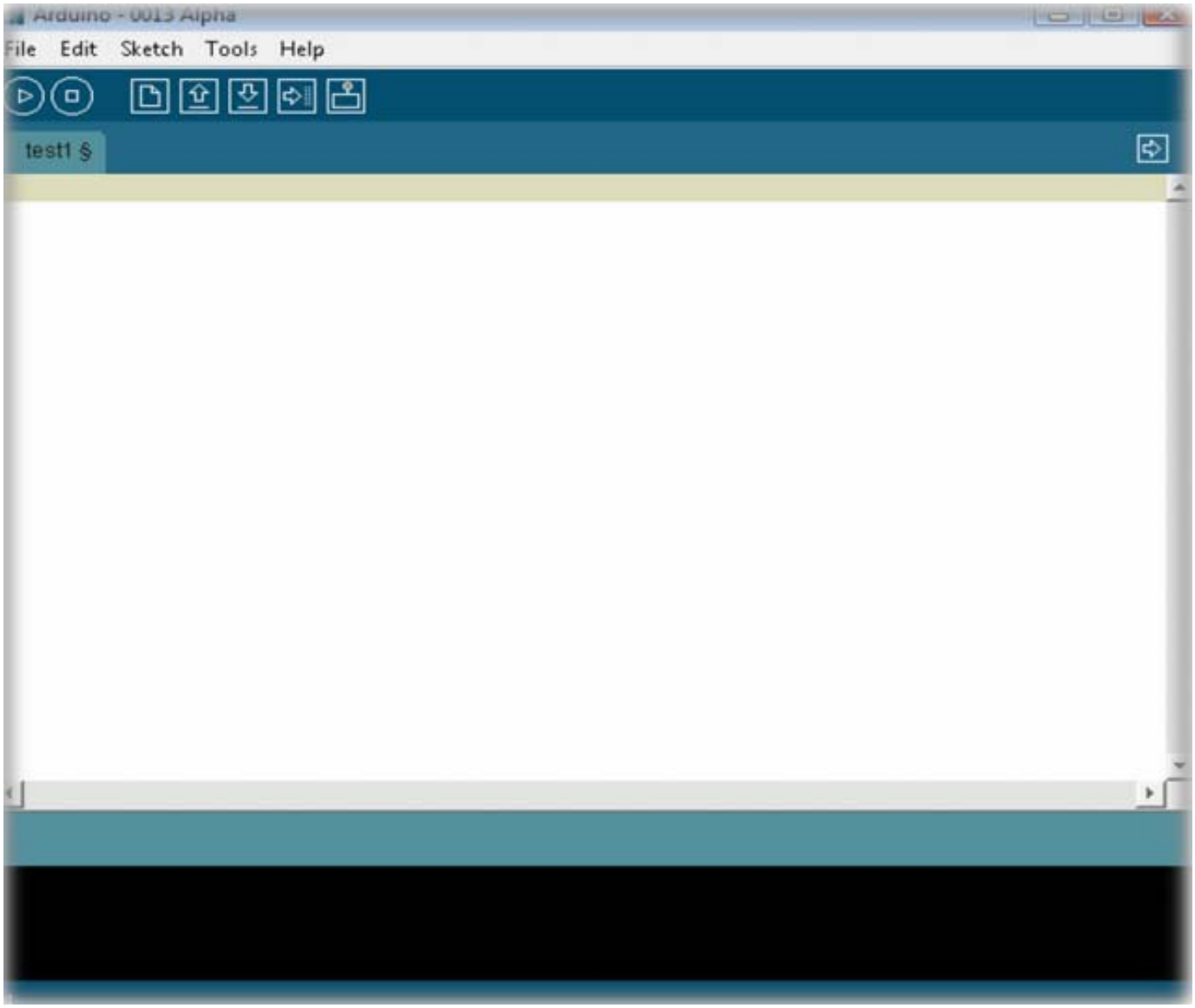
1.5.2 Προγραμματισμός εφαρμογών

Το προγραμματιστικό περιβάλλον που χρησιμοποιείται συνήθως στα παραδείγματα της πλατφόρμας ονομάζεται «processing». Είναι βασισμένο στην προγραμματιστική γλώσσα Java και έχει σχεδιαστεί για τη χρήση από σχεδιαστές και οποιονδήποτε άλλο που δεν χρειάζεται να γνωρίζει όλες τις λεπτομέρειες του προγραμματισμού σε χαμηλό επίπεδο μηχανής, αλλά επιθυμούν στην δημιουργία κάποιου «έργου». Είναι χρήσιμο εργαλείο για την πραγματοποίηση ιδεών προγραμματισμού επειδή απαιτείται σχετικά μικρός κώδικας επεξεργασίας για αρκετά σημαντικά πράγματα όπως η δημιουργία σύνδεσης δικτύων η σύνδεση μιας εξωτερικής συσκευής μέσω μιας σειριακής θύρας ή ο έλεγχος ψηφιακής κάμερας μέσω Fire Wire. Είναι ελεύθερο και ανοιχτό προς το κοινό εργαλείο, διαθέσιμο μέσο του διαδικτύου.

Σε περίπτωση που κάποιο κομμάτι της processing δεν είναι επιθυμητό είναι σε θέση να χρησιμοποιήσει δείγματα του κώδικα και τα κατάλληλα σχόλια με τη χρήση ψευδογλώσσας για οποιοδήποτε περιβάλλον πολυμέσων είναι επιθυμητό.

Η πλατφόρμα διαθέτει οκτώ κουμπιά επιλογών τα οποια είναι βεβαίωση / σύνταξη, διακοπή, νέο, άνοιγμα, αποθήκευση, μεταφόρτωση στην I/O πλατφόρμα, σειριακή οθόνη και αναγνωριστικός κατάλογος ή verify, stop, new, open, save, upload to I/O board and serial monitor. Το κουμπί επιλογής βεβαίωση / σύνταξη ελέγχει αν ο κώδικας του χρήση περιέχει λάθει και τα επισημαίνει ενώ το κουμπί διακοπή διακόπτει τη σειριακή επικοινωνία. Το κουμπί επιλογής νέο δημιουργεί νέα καρτέλα για την καταγραφή κώδικα, το κουμπί επιλογής άνοιγα παρουσιάζει όλους τους κώδικες που έχουν δημιουργηθεί και το κουμπί επιλογής αποθήκευση αποθηκεύει τον κώδικα. Το κουμπί επιλογής μεταφόρτωση στην I/O πλατφόρμα μεταβιβάζει τον κώδικα στην I/O πλατφόρμα αφού πρώτα έχει γίνει αποθήκευση και έλεγχος της σωστής σύνταξης του κώδικα. Το κουμπί επιλογής σειριακή οθόνη εμφανίζει τα σειριακά δεδομένα στην οθόνη τα οποία αποστέλλονται από την πλατφόρμα. Πρέπει να γίνει επιλογή της μεταβλητής ταχύτητας μεταβίβασης των δεδομένων του ηλεκτρονικού υπολογιστή ή αλλιώς των τηλεγραφικών σημάτων, σύμφωνα με τον κώδικα που έχει

συνταθεί. Το τελευταίο κουμπί του αναγνωριστικού καταλόγου επιτρέπει τη διαχείριση κωδικών με περισσότερα από ένα αρχεία. Τα αρχεία αυτά μπορεί να είναι αρχεία της γλώσσας του Arduino, της C,C++.



1.5.3 Σύνολο Εντολών

Τα arduino προγράμματα μπορούν να χωριστούν σε 3 κύρια μέρη : δομή (structure), τιμές (μεταβλητές και σταθερές), και οι εντολές (functions).

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi-line comment)
- #define
- #include

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)

- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)
- ! (not)

Pointer Access Operators

- * dereference operator
- & reference operator

Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- &= (compound bitwise and)
- |= (compound bitwise or)

Variables

Constants

- HIGH | LOW

- INPUT | OUTPUT| INPUT_PULLUP
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- short
- float
- double
- string - char array
- String - object
- array

Conversion

- char()
- byte()
- int()
- word()
- long()
- float()

Variable Scope & Qualifiers

- variable scope

- static
- volatile
- const

Utilities

- sizeof()

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Due only

- analogReadResolution()
- analogWriteResolution()

Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Time

- millis()
- micros()
- delay()
- delayMicroseconds()

Math

- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry

- sin()
- cos()
- tan()

Random Numbers

- randomSeed()
- random()

Bits and Bytes

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

External Interrupts

- attachInterrupt()
- detachInterrupt()

Interrupts

- interrupts()
- noInterrupts()

Communication

- Serial
- Stream

USB (Leonardo and Due only)

- Keyboard
- Mouse

1.5.4 Βιβλιοθήκες

Η πλατφόρμα Arduino συνοδεύεται μαζί με μερικές χρήσιμες βιβλιοθήκες όπου κάθε μία από αυτές παρέχει κάποια υπηρεσία είτε σε επίπεδο λογισμικού είτε σε επίπεδο υλικού. Επίσης, στο Διαδίκτυο μπορούν να βρεθούν και αρκετές βιβλιοθήκες όπου αναπτύσσονται κυρίως από προγραμματιστές που στηρίζουν το Arduino. Η χρήση αυτών σε οποιονδήποτε κώδικα γίνεται με την επιλογή «import library» ή χειροκίνητα με την εντολή #include (όνομα βιβλιοθήκης). Παρακάτω, ακολουθούν μερικές βασικές βιβλιοθήκες :

➢ PS2Keyboard

Ανάγνωση δεδομένων από πληκτρολόγια τύπου PS/2

➢ EEPROM

Διαχείριση της μνήμης EEPROM μιας πλατφόρμας Arduino

➢ Webduino

Επεκτάσιμος διαδικτυακός εξυπηρετητής ιστοσελίδων

➢ Ethernet

Ενσύρματη δικτυακή σύνδεση με την κάρτα επέκτασης Ethernet

➢ XBee

Ασύρματη δικτυακή σύνδεση με την κάρτα επέκτασης XBee

➢ LiquidCrystal

Διαχείριση τυπικών οθονών υγρών κρυστάλλων

➢ SD

Διαχείριση καρτών μνήμης SD

➢ Servo / Stepper

Διαχείριση και έλεγχος σερβοκινητήρων / βηματικών κινητήρων

➢ Finite State Machine

Υλοποίηση μηχανών πεπερασμένων καταστάσεων

➢ NewSoftSerial

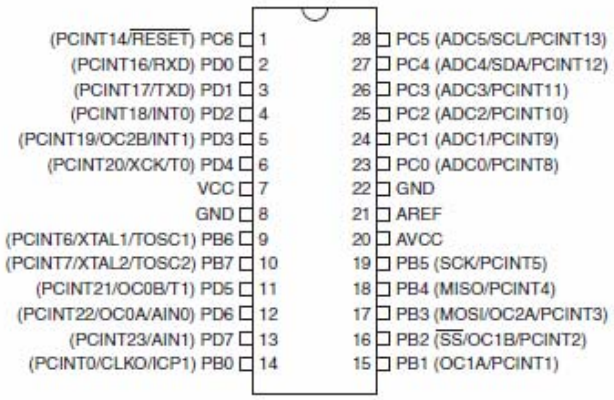
Υλοποίηση σειριακής επικοινωνίας σε επίπεδο λογισμικού

➢ Flash

Αποθήκευση δεδομένων στην μνήμη Flash μιας πλατφόρμας Arduino

1.6 Μικροελεγκτής Altem ATmega 328 8bit

Ο ATmega 328 είναι ολοκληρωμένο κύκλωμα, που ανήκει στην κατηγορία των μικροελεγκτών και περιέχει όλα τα απαραίτητα στοιχεία ενός ψηφιακού προγραμματιζόμενου συστήματος.
Το Arduino Uno έχει ενσωματωμένο αυτόν τον ελεγκτή και είναι η καρδιά του συστήματος μας.



1.6.1Σύνθεση των Pins

1.6.2 Περιγραφή των Pin

1.6.2.1 VCC

Ψηφιακή τάση τροφοδοσίας.

1.6.2.2 GND

Γείωση

1.6.2.3 Port B

Η PortB είναι μια αμφίδρομη θύρα εισόδου/εξόδου με εσωτερικές αντιστάσεις που έχουν επιλεγεί για κάθε bit. Οι καταχωρητές εξόδου της PortB έχουν συμμετρικά χαρακτηριστικά.

Ως είσοδοι τα pins της PortB έχουν χαμηλή τάση εάν οι αντιστάσεις έχουν ενεργοποιηθεί με υψηλή τάση. Τα pins της PortB είναι τριών καταστάσεων όταν η κατάσταση reset γίνει ενεργή και ο παλμός clock δεν είναι σε λειτουργιά . Ανάλογα με τις ρυθμίσεις του ρολογιού επιλογής, PB6 μπορεί να χρησιμοποιηθεί ως είσοδος στον ενισχυτή αναστροφής ταλαντωτή και ως εισόδους στο εσωτερικό κύκλωμα λειτουργίας ρολογιού.

Ανάλογα με τις ρυθμίσεις του ρολογιού επιλογής, PB7 μπορεί να χρησιμοποιηθεί ως έξοδος στον ενισχυτή αναστροφής ταλαντωτή.

1.6.2.4 PortC

Η PortC είναι μια 7 bit αμφίδρομη θύρα εισόδου/εξόδου με εσωτερικές αντιστάσεις που έχουν επιλεγεί για κάθε bit. .Οι καταχωρητές εξόδου της portc έχουν συμμετρικά χαρακτηριστικά. Ως είσοδοι τα pins της Portc έχουν χαμηλή τάση εάν οι αντιστάσεις έχουν ενεργοποιηθεί με υψηλή τάση. Τα pins της PortC είναι τριών καταστάσεων όταν η κατάσταση reset γίνει ενεργή και ο παλμός clock δεν είναι σε λειτουργιά

1.6.2.5 PortD

Η PortD είναι μια 8 bit αμφίδρομη θύρα εισόδου/εξόδου με εσωτερικές αντιστάσεις που εχουν επιλεγεί για κάθε bit. Οι καταχωρητές εξόδου της portd έχουν συμμετρικά χαρακτηριστικά. Ως είσοδοι τα pins της PortD έχουν χαμηλή τάση εάν οι αντιστάσεις έχουν ενεργοποιηθεί με υψηλή τάση. Τα pins της PortD είναι τριών καταστάσεων όταν η κατάσταση reset γίνει ενεργή και ο παλμός clock δεν είναι σε λειτουργιά

1.6.2.6 AVCC

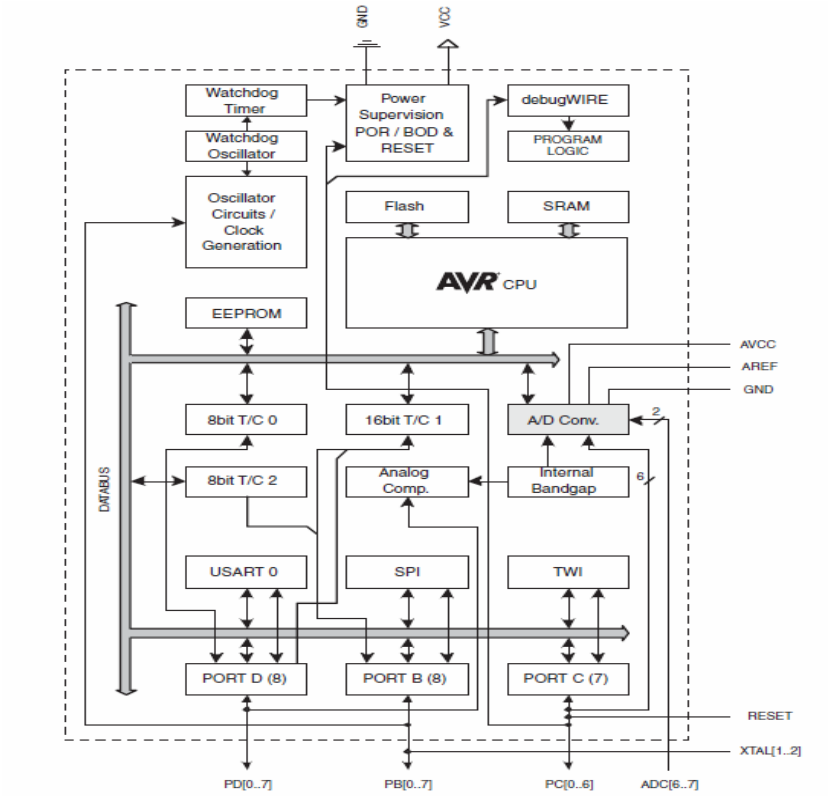
Η AV_{CC} είναι η τάση τροφοδοσίας για την μετατροπή αναλογικού σε ψηφιακό σήμα .Θα πρέπει να είναι και εξωτερικά συνδεδεμένη και με Vcc ακόμη και αν ο ADC δεν χρησιμοποιείται. Αν ο ADC χρησιμοποιείται, θα πρέπει να συνδεθεί με Vcc μέσω ενός βαθυπερατού φίλτρου.

1.6.3 Περιγραφή

Ο ATmega328 είναι ένας χαμηλής ισχύος CMOS 8-bit μικροελεγκτής με βάση το AVR σε ενισχυμένη αρχιτεκτονική RISC. Εκτελώντας ισχυρές διαδικασίες σε ένα μοναδικό κύκλο συγχρονισμού,ο ATmega328 επιτυγχάνει πολλαπλά MIPS ανά MHz που επιτρέπει το σύστημα σχεδιαστή να βελτιστοποιήσει την κατανάλωση ενέργειας σε σχέση με την ταχύτητα

επεξεργασίας του.

Block Διάγραμμα

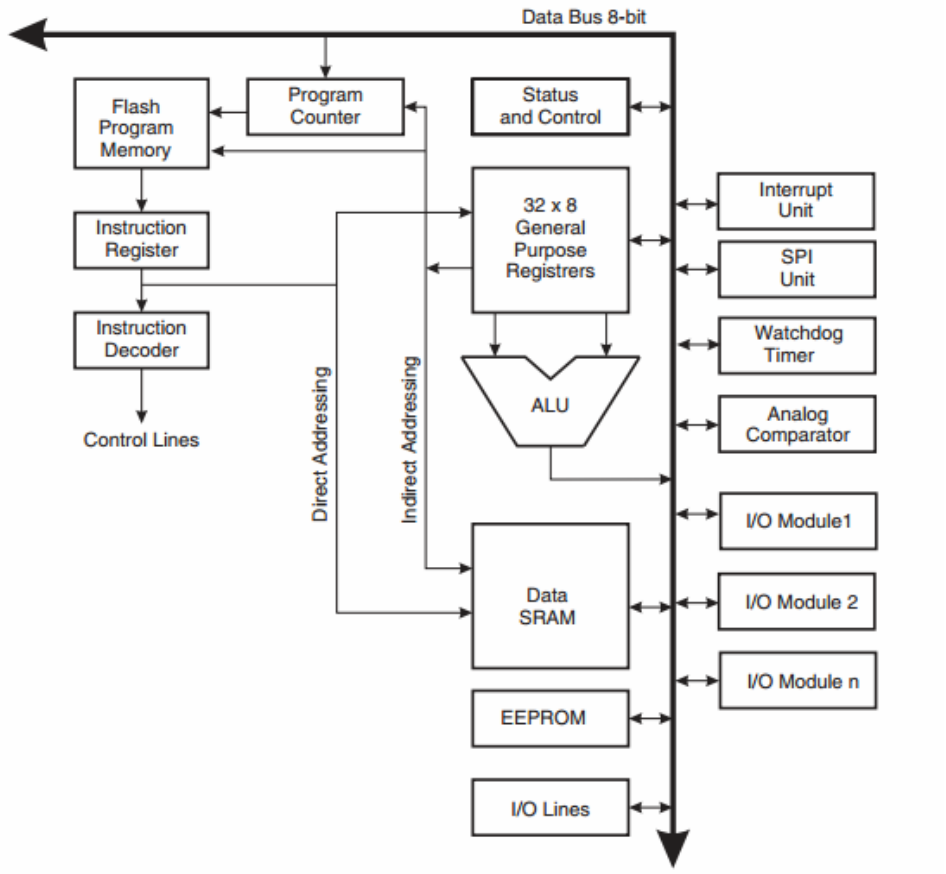


Ο πυρήνας γενικά ενός AVR επεξεργαστή και στην περίπτωση μας του ATmega328 συνδυάζει ένα πλούσιο σετ εντολών με 32 καταχωρητές εργασίας γενικού σκοπού. Και οι 3 καταχωρητές είναι άμεσα συνδεδεμένοι με την αριθμητική λογική μονάδα (ALU) επιτρέποντας δύο ανεξάρτητοι καταχωρητές να έχουν πρόσβαση σε μια απλή εντολή ενός κύκλου ρολογιού.Η δομή που προκύπτει έχει ποιο αποτελεσματικό κώδικα επιτυγχάνοντας παράλληλα ταχύτατα επεξεργασίας έως και 10 φορές μεγαλύτερη από έναν συμβατικό μικροελεκτή τύπου CISC.

Ο ATmega 328 παρέχει τις ακόλουθες λειτουργίες :

32Kbytes of In-System Programmable Flash με δυνατότητα ανάγνωσης-εγγραφής, 1Kbytes EEPROM, 2Kbytes SRAM, 23 γραμμές γενικής χρήσης για είσοδο-έξοδο, 32 καταχωριτές εργασίας γενικού σκοπού, 3 απαριθμητές/χρόνισες, εσωτερικές και εξωτερικές διακοπές, μια σειριακά προγραμματιζόμενη USART, μια SPI σειριακή θύρα, έναν 6καναλο 10bit DAC, χρονιστή εσωτερικά προγραμματιζόμενο με εσωτερικό κρύσταλλο.

Block διάγραμμα εσωτερικά ενός Avr επεξεργαστή



Με στόχο να αυξήσουν την απόδοση τους οι οικογένεια επεξεργαστών AVR και κατά συνέπεια και ο ATmega 328 βασίζονται και κάνουν χρήση της αρχιτεκτονικής Harvard με ξεχωριστές θέσεις μνήμης και διαύλους για το πρόγραμμα και τα δεδομένα . Οι οδηγίες στη μνήμη προγράμματος εκτελούνται με ένα μόνο επίπεδο αγωγών. Όταν μια εντολή εκτελείται, η επόμενη εντολή προ-ανασύρεται από τη μνήμη προγράμματος .Αύτη η σχεδίαση επιτρέπει κάθε εντολή να εκτελείτε σε έναν κύκλο ρολογιού .

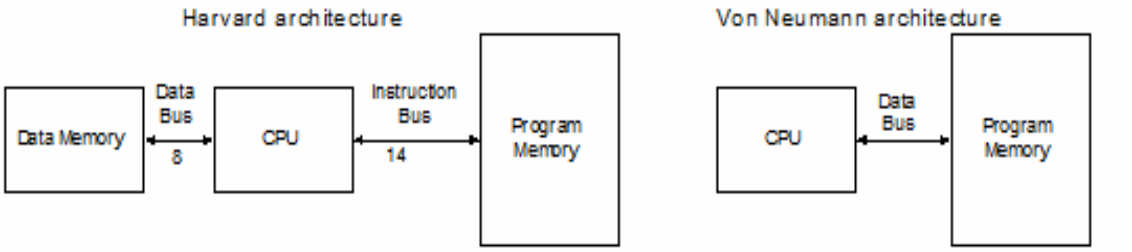
1.6.3.1 Αρχιτεκτονικές Harvard και Von-Neumann

Οι αρχιτεκτονικές Harvard και Von-Neumann αποτελούν τις δύο βασικές αρχιτεκτονικές των σύγχρονων μικροϋπολογιστικών συστημάτων. Οι μικροελεγκτές που είναι σχεδιασμένοι με βάση την αρχιτεκτονική Harvard καλούνται επίσης και μικροελεγκτές RISC (Reduced Instruction Set Computer) ενώ εκείνοι που χρησιμοποιούν την αρχιτεκτονική Von-Neumann καλούνται μικροελεγκτές CISI (Complex Instruction Set Computer).

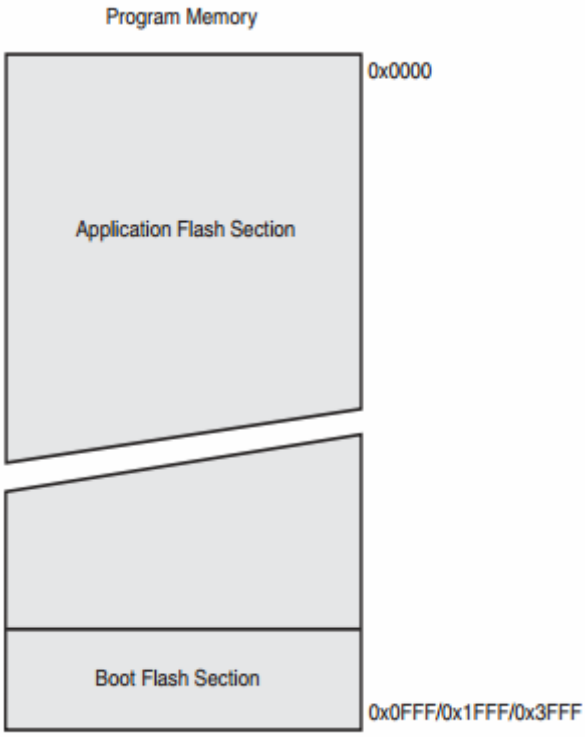
Όπως εύκολα μπορεί κανείς να παρατηρήσει και από το σχήμα, στην αρχιτεκτονική Harvard υπάρχει διαφορετικός δίαυλος για τη μεταφορά δεδομένων (data bus) και διαφορετικός δίαυλος για τη μεταφορά των εντολών (instruction bus). Η ύπαρξη δύο διαφορετικών μνημών, μνήμη

δεδομένων (data memory) και μνήμη προγράμματος (program memory), καθιστά την αρχιτεκτονική Harvard πιο αποδοτική, αφού μπορεί να εκτελείται κάποια εντολή και παράλληλα να εγγράφεται ή να διαβάζεται η μνήμη. Με τον τρόπο αυτό επιτυγχάνεται η εκτέλεση της εντολής σε ένα μόνο χρόνο μηχανής.

Επίσης, η αρχιτεκτονική Harvard επιτρέπει οι εντολές να έχουν διαφορετικό μήκος σε δυαδικά ψηφία (binary digits, bit) από τα δεδομένα. Δίνεται η δυνατότητα να επιλέγεται, ανάλογα με το πλήθος των εντολών, το κατάλληλο μήκος της λέξης εντολής ώστε να επιτευχθεί η κωδικοποίηση της κάθε εντολής σε μία μόνο λέξη. Πετυχαίνεται με αυτό τον τρόπο να μειωθεί σημαντικά η ταχύτητα ανάκλησης (fetch) της κάθε εντολής. Είναι τέλος ενδεικτικό της αρχιτεκτονικής αυτής ο μειωμένος αριθμός εντολών καθώς και η εκτέλεση της κάθε εντολής σε ένα μόνο χρόνο μηχανής.



1.6.4 Μνήμη ATmega328



1.6.4.1 SRAM DataMemory

32 Registers	0x0000-0x001F
64 I/O Registers	0x0020-0x005F
160 EXT I/O Registers	0x0060-0x00FF
Internal SRAM 2048	0x08FF

Ο ATmega328 είναι ένα σύστημα μικροεπεξεργαστή με περιφερειακές μονάδες οι οποίες μπορούν να υποστηρίζονται στο πλαίσιο των 64 θέσεων που ορίζει το Opcode για είσοδο και έξοδο. Για εκτεταμένη είσοδο/έξοδο παρέχει χώρο από την θέση 0x60-0xFF μέσα στην SRAM. Τα 2303 χαμηλότερα Bytes θέσεις μνήμης αποτελούν ταυτόχρονα και το αρχείο των καταχωριτών, την μνήμη εισόδου/εξόδου, εκτεταμένη μνήμη εισόδου/εξόδου και τα εσωτερικά δεδομένα της SRAM . Οι πρώτες 32 θέσεις περιέχουν το αρχείο των καταχωρητών οι επόμενες 64 την στάνταρ είσοδο /έξοδο οι επόμενες 160 την εκτεταμένη είσοδο/έξοδο και τέλος οι επόμενες 2048 τα εσωτερικά δεδομένα της SRAM .

1.6.5 Σύγκριση ATmega328 με άλλους τις ίδιας σειράς

Device	**Flash**	**EEPROM**	**RAM**	**Intweeupt Vector Size**
ATmega48A	4Kbytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4Kbytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8Kbytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8Kbytes	512Bytes	1KBytes	1 instruction word/vector

ATmega168A	16Kbytes	512Bytes	1KBytes	2 instruction word/vector
ATmega168PA	16Kbytes	512Bytes	1KBytes	2 instruction word/vector
ATmega328A	32Kbytes	1Kbytes	2KBytes	2 instruction word/vector
ATmega328PA	32Kbytes	1Kbytes	2KBytes	2 instruction word/vector

ΚΕΦΑΛΑΙΟ 2ο

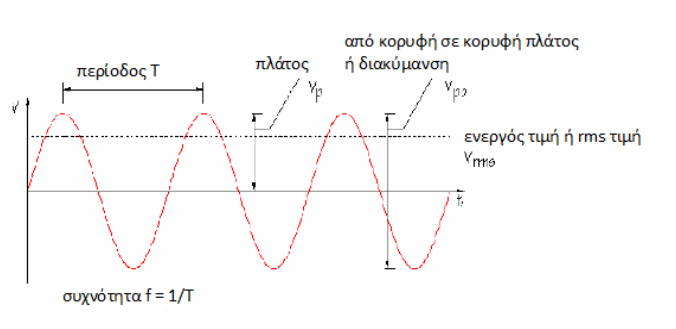
2.1 Γεννήτρια σημάτων

Οι γεννήτριες σημάτων είναι όργανα που παρέχουν σήμα συνήθως με τη μορφή τάσης ή ρεύματος στο κύκλωμα. Έχουν την δυνατότητα να παράγουν σήματα με ποικίλα χαρακτηριστικά τα οποία ελέγχονται από τον χρήστη. Τα σήματα που παράγονται από μια γεννήτρια, άλλα είναι επαναλαμβανόμενα και άλλα μη επαναλαμβανόμενα (αναλογικά και ψηφιακά). Στην κατηγορία αυτή ανήκουν και οι γεννήτριες συχνοτήτων, οι οποίες και παράγουν επαναλαμβανόμενες κυματομορφές τις οποίες από την ψηφιακή τους μορφή τις μετατρέπουμε σε αναλογική και η εποπτεία τους γίνεται με την βοήθεια παλμογράφου. Οι πιο σύνηθες επαναλαμβανόμενες κυματομορφές είναι η ημιτονική, τριγωνική, πριονωτή και τετραγωνική.

2.2 Συνεχές ρεύμα (AC) και Εναλλασσόμενο ρεύμα (DC)

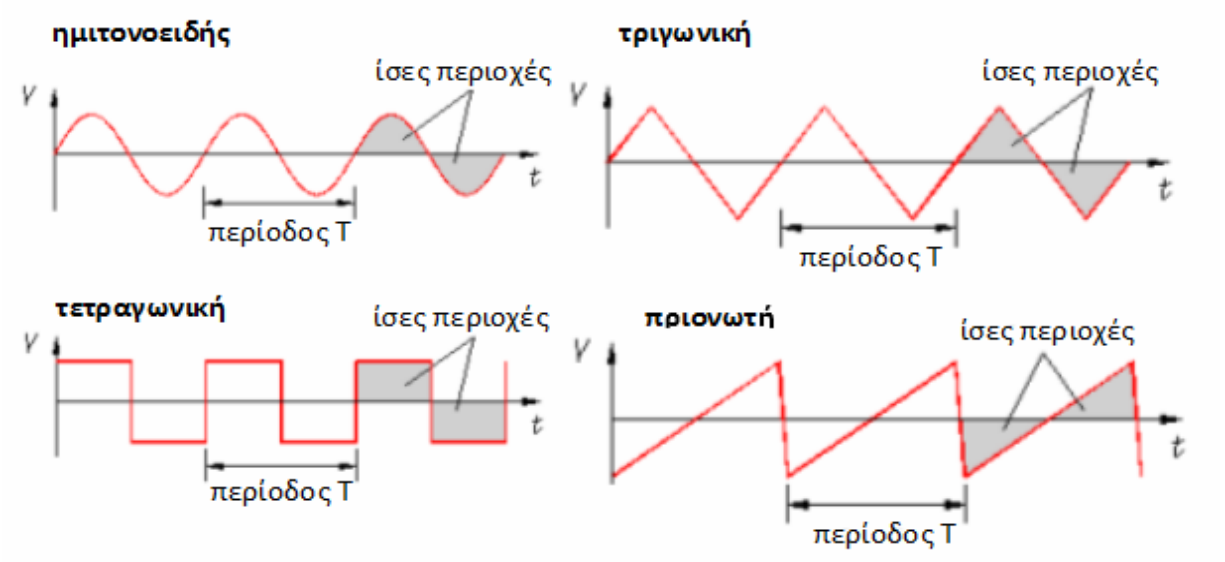
Το συνεχές ρεύμα είναι η σταθερή ροή ηλεκτρονίων προς μια κατεύθυνση. Γενικά οι συνεχείς τάσεις παράγονται από τροφοδοτικά, μπαταριές, γεννήτριες. Μια συνεχής τάση έχει ένα συγκεκριμένο πλάτος και μια συγκεκριμένη κατεύθυνση που την καθορίζει.
Τα τροφοδοτικά συνεχούς ρεύματος τάσης δεν αλλάζουν την τιμή τους με την πάροδο του χρόνου και έχουν μια σταθερή τιμή. Άρα το συνεχές ρεύμα (DC) κρατά σταθερή τιμή για πάντα.

Μια κυματομορφή εναλλασσόμενου ρεύματος (AC) από την άλλη μεριά ορίζεται ως εκείνη που μεταβάλλεται και στο μέγεθος και στην κατεύθυνση με έναν ομαλό τρόπο ως προς τον χρόνο .Ο Όρος εναλλασσόμενο ρεύμα γενικά αναφέρεται σε κυματομορφές που μεταβάλλονται με τον χρόνο με παράδειγμα αυτές που δημιουργήσαμε στην κατασκευή της γεννήτριας με πιο γνωστή την ημιτονική κυματομορφή.



2.3 Είδη σημάτων - Γενικά στοιχεία

Όπως σε όλα τα σήματα, έτσι και στα ηλεκτρικά, η τάση και ένταση του ηλεκτρικού ρεύματος, διακρίνεται σε περιοδική και μη. Περιοδικά είναι τα σήματα που επαναλαμβάνονται ανά τακτά χρονικά διαστήματα. Στα περιοδικά σήματα η μεταβολή της τάσης ακολουθεί έναν κύκλο, δηλαδή επαναλαμβάνεται με συγκεκριμένο τρόπο σε συγκεκριμένο χρονικό διάστημα. Το χρονικό διάστημα που απαιτείται για να ολοκληρώσει μία πλήρης μεταβολή ονομάζεται περίοδος του σήματος και συνήθως συμβολίζεται με T. Ο αριθμός των επαναλαμβανόμενων μεταβολών σε ένα δευτερόλεπτο ονομάζεται συχνότητα και συνήθως συμβολίζεται με f (frequency). Ισχύει f=1/T.
Το πλάτος A είναι το μέγεθος ή η ένταση του κύματος της κυματομορφής και μετριέται σε volts. Η χρονική απόσταση μεταξύ δύο τμημάτων του αυτού σήματος ή δύο διαφορετικών σημάτων ίδιας περιόδου (από τα οποία το ένα θεωρείται ως αναφορά) ονομάζεται διαφορά φάσης. Τα μη περιοδικά σήματα είναι καταστάσεις οι οποίες συμβαίνουν σε ακανόνιστα χρονικά διαστήματα. Σε αυτά τα σήματα μπορούμε να μιλήσουμε μόνο για χρονική διάρκεια και μορφή. Η ονομασία κάθε κυματομορφής προέρχεται είτε από τη συνάρτηση (μαθηματική έκφραση) η οποία το περιγράφει, είτε από την περιγραφή της μορφής του σήματος, όπως τριγωνικό, τετραγωνικό, πριονωτό κλπ.



2.3.1 Ημιτονικό Σήμα

Το ημιτονικό σήμα, όπως αναφέραμε είναι ένα περιοδικό σήμα, στο οποίο η μεταβολή της τάσης, μεταβάλλεται ανάλογα με τις τιμές που παίρνει το ημίτονο. Η μαθηματική έκφραση που περιγράφει το ημιτονικό σήμα είναι $V=V_0$ημ(ωt), όπου V_0 το πλάτος της κυματομορφής στο οποίο η συνάρτηση παρουσιάσει μέγιστο, ω η γωνιακή ταχύτητα (ισοδυναμεί με 2πf, όπου π=3.14, f η συχνότητα) και t η χρονική στιγμή, που μετριέται από την αρχή των χρόνων που ορίζουμε

2.3.2 Τετραγωνικό Σήμα

Το τετραγωνικό σήμα είναι περιοδικό, στο οποίο η τάση έχει δύο τιμές, την ελάχιστη (ή LOW level) και τη μέγιστη (ή HIGH level). Στη γραμμή λοιπόν εμφανίζεται αυτή η συνεχής εναλλαγή των δύο τιμών. Ένα άλλο χαρακτηριστικό μέγεθος του τετραγωνικού σήματος είναι το duty cycle, το οποίο δηλώνει το ποσοστό επί της % της διάρκειας του HIGH level σε σχέση με την περίοδο.
Ένα τετραγωνικό σήμα δεν είναι τίποτα παραπάνω από ένα άθροισμα από ένα άπειρο αριθμό από ημιτονικά σήματα το καθένα με μια συχνότητα που είναι περιττό πολλαπλάσιο της θεμελιώδους συχνότητας. Αν κάνουμε ένα γράφημα με αυτά τα σήματα και τα προσθέταμε μεταξύ τους σημείο προς σημείο θα παίρναμε το τετραγωνικό σήμα.

2.3.3 Τριγωνικό Σήμα

Το σχήμα ενός τριγωνικού σήματος είναι πολύ κοντά στο σχήμα ενός ημιτονικού σήματος σε αντίθεση με το τετραγωνικό σήμα, το οποίο διαφέρει αρκετά ειδικά στο ότι περιέχει ασυνέχειες δηλαδή χρονικές στιγμές στις οποίες η τιμή της τάσης μεταβάλλεται απότομα. Έτσι, τα κυκλώματα παραγωγής τριγωνικού σήματος συνοδεύονται απαραίτητα από παραγωγή τετραγωνικού.

2.4 Μέτρηση στο πεδίο της συχνότητας

Όλα τα παραπάνω αναφέρονται στο πεδίο του χρόνου. Αλλά τα σήματα μπορούν να αναπαρασταθούν και στο πεδίο της συχνότητας. Το πεδίο της συχνότητας αποτελεί μια διαφορετική οπτική γωνία για να μελετάμε τα σήματα. Ένας παλμογράφος απεικονίζει τα σήματα στο πεδίο του χρόνου το οποίο είναι χρήσιμο για ένα μεγάλο πλήθος μετρήσεων. Αλλά πολλές φορές μας ενδιαφέρουν η συχνότητα ή οι συχνότητες που έχει μια κυματομορφή. Το όργανο για την μεταφορά μας στο πεδίο συχνοτήτων ονομάζεται αναλυτής φάσματος .

2.5 Υλοποίηση των σημάτων με το Arduino

Το Arduino έχει αναλογικά και ψηφιακά pin, το πρόβλημα είναι ότι ενώ κάποια από τα σήματα που ζητούνται είναι αναλογικά το arduino δεν μπορεί να δώσει έξοδο αναλογικό σήμα αλλά μόνο ψηφιακό. Έτσι πρέπει τα σήματα να γίνουν με τα ψηφιακά Pin και να περάσουν από έναν μετατροπέα ψηφιακό σε αναλογικό. Στην παρούσα εργασία ο μετατροπέας είναι ένα δικτύωμα αντιστάσεων R-2R το οποίο αναλύεται εκτενώς παρακάτω (κεφ. 4.2). Σκοπός είναι να κατασκευαστούν τα 4 σήματα που ζητούνται με την χρήση προσεγγιστικών μεθόδων. Στην παρούσα εργασία χρησιμοποιούμε τα πρώτα 8 pin(0 – 7) του board έτσι μπορούμε να έχουμε 2^8 = 256 τιμές πλάτους.

2.5.1 Τριγωνικό σήμα

Το τριγωνικό σήμα μαθηματικά είναι μία συνεχής συνάρτηση στον χρόνο η οποία ξεκινώντας από το μέσο πλάτος αυξάνει μέχρι ένα ανώτατο πλάτος και κατεβαίνει στο κατώτατο με σταθερό ρυθμό. Αυτό προγραμματιστικά μπορεί να υλοποιηθεί με έναν βρόχο ο οποίος θα αυξάνει μία μεταβλητή κατά 1 μέχρι η μεταβλητή να πάρει την μέγιστη προσεγγιστική τιμή που έχουμε(256) και θα την δίνει στην έξοδο. Μόλις φτάσει σε αυτήν την τιμή θα μειώνει μέχρι να φτάσει στην κατώτατη τιμή που είναι το 0. Μόλις φτάσει στο 0 αρχίζει να αυξάνει πάλι και η διαδικασία επαναλαμβάνεται συνεχώς. Η ακέραια τιμή της μεταβλητής βρόχου ανατίθεται στον καταχωρητή PORTD ο οποίος μετατρέπει την τιμή στο δυαδικό και δίνει τάση high (5 volt) στις αντίστοιχες εξόδους των pin. Για παράδειγμα εάν ο PORTD πάρει την τιμή 127(11110000) τότε θα έχουν τάση τα πρώτα 4 pin(4-7) ενώ τα άλλα 4(0 – 3) θα είναι σε κατάσταση low. Έτσι εφόσον αυτός ο βρόχος εκτελείται συνεχώς στην έξοδο έχουμε ένα τριγωνικό σήμα.

2.5.2 Ημιτονικό σήμα

Το ημιτονικό σήμα είναι το δυσκολότερο από τα 4 σήματα καθώς δεν αρκεί να έχουμε μία μεταβλητή που αυξάνει διότι το ημίτονο έχει συγκεκριμένες τιμές κάθε στιγμή. Έτσι γίνεται χρήση ενός βοηθητικού πίνακα 255 θέσεων ο οποίος παίρνει τιμές ημιτόνου σε ένα εύρος τιμών από 0 εώς 255,έτσι ώστε να εμφανίζεται στην έξοδο ένα σήμα που προσεγγίζει το ημιτονικό σήμα. Αφού έχουμε τον πίνακα τιμών τότε η διαδικασία είναι ίδια με αυτή του τριγωνικού σήματος με μόνη διαφορά ότι αντί ο PORTD να παίρνει τις τιμές της μεταβλητής του βρόχου, παίρνει τις τιμές του πίνακα και η μεταβλητή του βρόχου χρησιμοποιείται ως δείκτης αύξησης τιμών του πίνακα.

2.5.3 Τετραγωνικός παλμός

Ο τετραγωνικός παλμός υλοποιείται ιδιαίτερα εύκολα αρκεί να δίνουμε στην έξοδο 0 για μία περίοδο και 255 για την επόμενη συνεχόμενα.

2.5.4 Πριονωτό σήμα

Θα μπορούσαμε να πούμε πως το πριονωτό σήμα είναι μία υποπερίπτωση του τριγωνικού σήματος καθώς η μόνη τους διαφορά είναι ότι το πριονωτό σήμα αφού φτάσει στην ανώτατη στάθμη τάσης αντί να μειώνεται μηδενίζεται. Αυτό γίνεται με την χρήση ενός βρόχου for ο οποίος αυξάνει συνεχώς μία μεταβλητή i από το 0 μέχρι το 255 της οποίας η τιμή ανατίθεται στον καταχωρητή PORTD. Έτσι έχουμε ένα πριονωτό σήμα στην έξοδο.

2.6 Τα σήματα στο πεδίο των συχνοτήτων

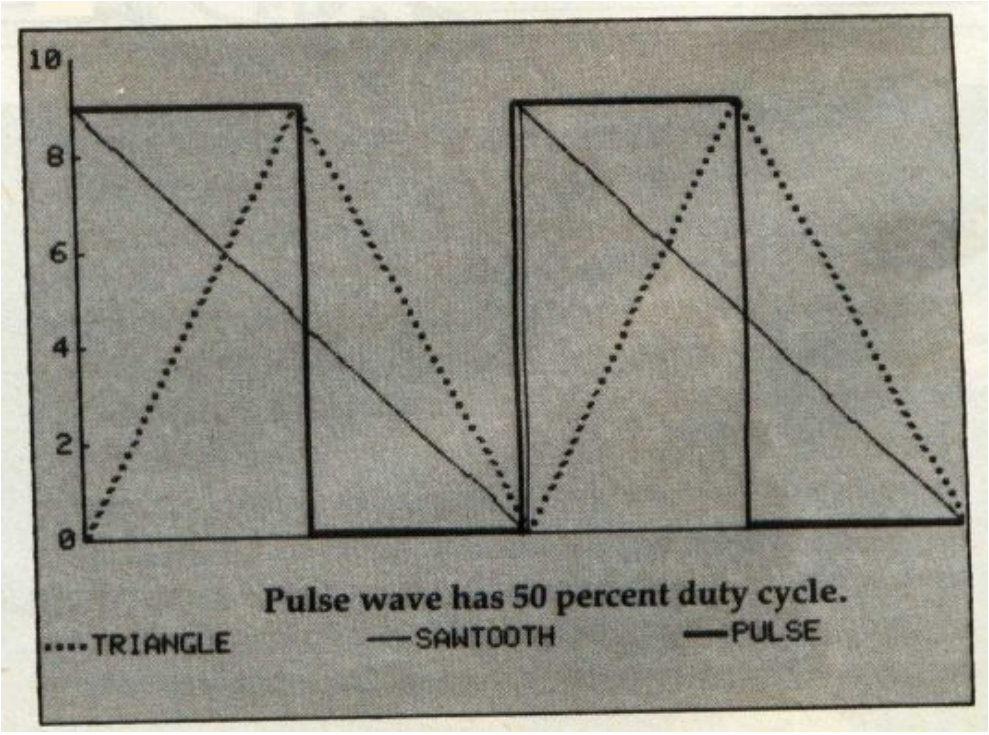
Μαθηματικά με την ανάπτυξη των σειρών fourier για τα σήματα της εργασίας μπορούμε να δούμε την συμπεριφορά τους στο πεδίο των συχνοτήτων.

Έστω $x = 2\pi ft$

Τετραγωνικό-σήμα: $h(t) = 4\{(\sin(x/1) + (\sin(3x/3)) + (\sin(5x/5) + ...\}$

Τριγωνικό-σήμα: $h(\tau) = (\pi/2) - (4/\pi)\{(\cos(x) + \cos(x/3) + (\cos(x/5) + ...\}$

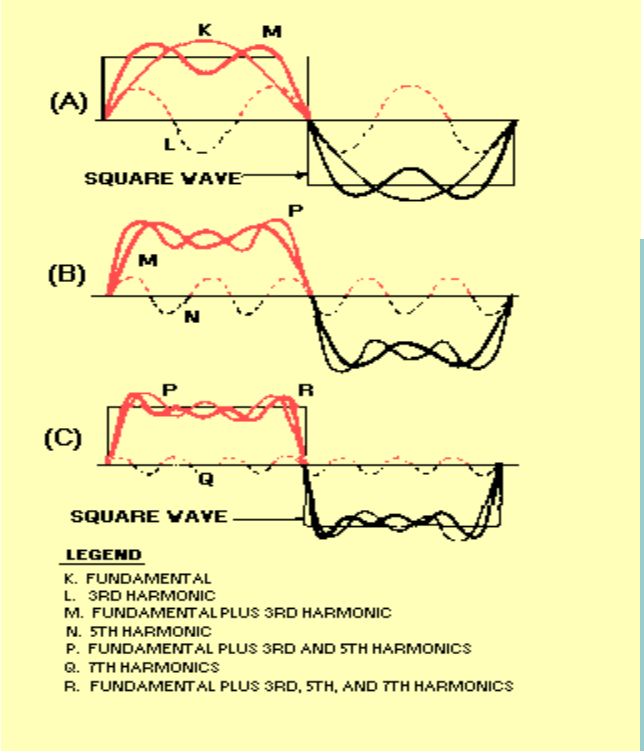
Πριονωτό-σήμα: $h(t) = 2\{(\sin(x) - (\sin(2x/2) + (\sin(3x/3) - (\sin(4x/4) + (...) - (...)\}$

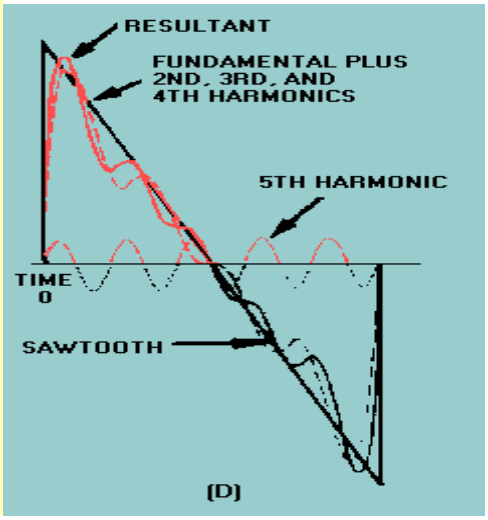


Η συμμετρία της κυματομορφής του τετραγωνικού και του πριονωτού σήματος ως προς τον άξονα του χρόνου, φανερώνει ότι αυτά δεν περιλαμβάνουν συνεχή συνιστώσα.
Τα ημιτονοειδή είναι απλές κυματομορφές καθαρών τόνων (ημιτονικά σήματα) από το συνδυασμό των οποίων προκύπτουν σύνθετα κύματα. Προσθέτοντας επιλεγμένους ως προς το συχνοτικό ύψος καθαρούς τόνους έχοντες την κατάλληλη ενεργειακή στάθμη-καθοριζόμενη από τη σχετική ανάλυση Fourier-και φάση αναμεταξύ τους μπορούν να δημιουργηθούν τετράγωνα, τριγωνικά και πριονωτά κύματα όπως και ευρεία ποικιλία μη ημιτονοειδών κυματομορφών. Η συγκεκριμένη Αρχή "Προσθετικής Σύνθεσης" εφαρμόζεται και από τα ηχοσυστήματά μας. Ο ήχος εκάστοτε ακουστικού οργάνου μιας ορχήστρας συνίσταται από μια θεμελιώδη και σειρά αρμονικών(με συχνοτικά ύψη ακέραια πολλαπλάσια της θεμελιώδους) και προκύπτει από συνολική υπέρθεση των επιμέρους κυματομορφών.

Το φάσμα των σύνθετων κυματομορφών περιλαμβάνει τη θεμελιώδη και αρμονικές(ακέραια πολλαπλάσια) συχνοτικές συνιστώσες, με το σύνθετο σχήμα να καθορίζεται από το πλήθος, τον τύπο τα σχετικά πλάτη και φάση αυτών. Όσο μεγαλύτερη η κλίση των σκελών της, τόσο μεγαλύτερο το πλήθος των αρμονικών που συνθέτουν την κυματομορφή. Στο επόμενο παράδειγμα, η προσθήκη εκάστης ανώτερης αρμονικής αυξάνει το πλάτος της συνισταμένης και κάνει τα σκέλη της πιο επικλινή. Οι κυματομορφές της 3ης, 7ης, 11ης, κ.ο.κ. αρμονικών συνιστωσών διέρχονται από το μηδενικό σημείο αναφοράς στο τέλος της περιόδου με διαφορά φάσης 180 μοιρών(ανάστροφη φάση) σε σχέση με τη θεμελιώδη, ενώ οι αντίστοιχες της 5ης, 9ης, 13ης,κ.ο.κ. το διασχίζουν σε φάση με τη θεμελιώδη.

Στο παρακάτω γράφημα, απεικονίζεται τετραγωνικό σήμα(μη ημιτονοειδές κι εναλασσόμενο εξ ορισμού μεταξύ δύο τιμών μηδενικού και πλάτους κορυφής) μαζί με το ημιτονοειδές K που αντιστοιχεί στην ίδια(θεμελιώδη) συχνότητα με το πρώτο. Με την υπέρθεση(και τη συνεπαγόμενη αλγεβρική προσθήκη των αντιστοίχων στιγμιαίων τιμών) του ημιτονοειδούς L που βρίσκεται σε τριπλάσιο τονικό ύψος(τρίτη αρμονική) και στο ένα τρίτον του πλάτους από το K, προκύπτει η κυματομορφή M. Προσθέτοντας ακολούθως την πέμπτη αρμονική συνιστώσα N (με πλάτος το ένα πέμπτον της K) προκύπτει η νέα καμπύλη P ενώ με την αντίστοιχη προσθήκη της εβδόμης αρμονικής Q (πλάτους 1/7 της K) καταλήγουμε στην καμπύλη R. Όσο αυξάνει το πλήθος των περιττών αρμονικών που υπερτίθενται(με τις ανάλογες ταπεινώσεις στο πλάτος),τόσο πιστότερα προσεγγίζει η προκύπτουσα σύνθετη κυματομορφή το σχήμα του τέλειου τετραγωνικού παλμού-ο οποίος συντίθεται από άπειρο πλήθος περιττών αρμονικών. Κατά τη σύνθεση τετραγωνικών σημάτων, οι περιττές αρμονικές συνιστώσες στο σύνολό τους διασχίζουν το σημείο αναφοράς ΄0΄ ούσες στην ίδια φάση με τη θεμελιώδη. Εν κατακλείδι, το τετραγωνικό σήμα αποτελείται από το ημιτονικό σήμα της θεμελιώδους καθώς και από εκείνα των αρμονικών με συχνοτικό ύψος 3πλάσιο,5πλάσιο,7πλάσιο,κ.ο.κ. της θεμελιώδους και πλάτος που βαίνει συνεχώς μειούμενο αντίστροφα προς την τάξη της αρμονικής.





ΚΕΦΑΛΑΙΟ 3ο

3.1 Γενικά για τους DAC

Οι μετατροπείς πληροφοριών (κυκλώματα που μετατρέπουν το αναλογικό σήμα σε ψηφιακό και αντίστροφα) διαδραματίζουν ένα πολύ σημαντικό ρόλο στη σύγχρονη ψηφιακή εποχή. Ποσότητες όπως η τάση, το ρεύμα, ο χρόνος, το φορτίο, η θερμοκρασία, η πίεση εμφανίζονται σε αναλογική μορφή. Για την επεξεργασία, τη μεταφορά και την αποθήκευση των πληροφοριών που μεταφέρουν οι παραπάνω μεταβλητές, χρειάζεται η μετατροπή τους σε ψηφιακή μορφή. Γνωρίζουμε ότι ένα σήμα μεταφέρει πληροφορία. Έτσι, το κύκλωμα που μετατρέπει ένα αναλογικό σήμα σε ψηφιακό ονομάζεται Αναλογικός-Ψηφιακός μετατροπέας (Analog to Digital Converter ή ADC). Το κύκλωμα που εκτελεί την αντίστροφη λειτουργία, δηλαδή μετατρέπει ένα ψηφιακό σήμα (συνήθως δυαδικό) σε αναλογικό ονομάζεται Ψηφιακός-Αναλογικός μετατροπέας (Digital to Analog Converter ή DAC)

Οι DAC μετατροπείς βρίσκουν σήμερα πολλές εφαρμογές. Χρησιμοποιούνται για την παραγωγή ηχητικών σημάτων από ψηφιακή πληροφορία σε συσκευές αναπαραγωγής μουσικής (CD-DVD-MP3). Οι ψηφιακές πληροφορίες που είναι αποθηκευμένες σε CD/DVD μετατρέπονται σε ήχο με τη βοήθεια ενός υψηλής ακρίβειας DAC. Το ίδιο συμβαίνει και με τα αρχεία μορφής MP3 που έχουμε αποθηκευμένα στον υπολογιστή μας και μπορούμε να ακούσουμε από τα ηχεία μέσω ενός DAC που βρίσκεται στη κάρτα ήχου του υπολογιστή. Τα ψηφιακά σήματα μετατρέπονται σε αναλογικά σε τηλεοράσεις και κινητά τηλέφωνα, δίνοντας πληροφορίες ώστε να εμφανίζονται τα χρώματα και οι αποχρώσεις στην οθόνη. Μέχρι το 2007, οι αναλογικές είσοδοι χρησιμοποιούνταν πιο συχνά από τις ψηφιακές, αλλά αυτό άλλαξε όταν επίπεδες οθόνες με DVI και συνδέσεις HDMI έγιναν πιο διαδεδομένες. Ένας βίντεο DAC ενσωματώνεται σε κάθε ψηφιακή συσκευή αναπαραγωγής βίντεο με αναλογικές εξόδους. Ο DAC συνήθως συνδυάζεται με κάποια μνήμη (RAM), η οποία περιέχει πίνακες μετατροπής για αντίθεση και φωτεινότητα και αποτελεί μια συσκευή που ονομάζεται RAMDAC. Σε εφαρμογές VoIP (Voice over IP), οι πληροφορίες που φτάνουν στο κινητό τηλέφωνο του δέκτη με τη μορφή ηλεκτρομαγνητικού κύματος (αναλογικό σήμα) μετατρέπονται μέσω ενός ADC σε ψηφιακή μορφή, επεξεργάζονται (ώστε ναφέρουμε το σήμα σε συχνότητες που είναι δυνατόν να γίνουν αντιληπτές από το ανθρώπινο αυτί) και στη συνέχεια με τη βοήθεια ενός DAC το σήμα μετατρέπεται από ψηφιακό σε αναλογικό που είναι πια η γνωστή μας φωνή που ακούμε στο ακουστικό. Μια ανάλογη διαδικασία πραγματοποιείται και από τη πλευρά του πομπού. Το αναλογικό σήμα που παράγεται από το πομπό (φωνή) μμεταφέρεται σε έναν ADC που το μμετατρέπει σε ψηφιακό. Το ψηφιακό σήμα υφίσταται κάποια

επεξεργασία και στη συνέχεια μέσω ενός DAC επανέρχεται σε αναλογική μορφή (ηλεκτρομαγνητικό κύμα) και είναι έτοιμο να σταλεί μέσω της κεραίας στο δίκτυο
που αναλαμβάνει την επικοινωνία πομπού -δέκτη.

Σήμερα λόγω του κόστους και κατ' επέκταση της ανάγκης για κυκλώματα που καταλαμβάνουν μμικρότερη επιφάνεια, οι DAC κατασκευάζονται σχεδόν αποκλειστικά σε ολοκληρωμένα κυκλώματα (Integrated Circuits). Υπάρχουν πολλές αρχιτεκτονικές DAC που έχουν διαφορετικά πλεονεκτήματα και μμειονεκτήματα. Η καταλληλότητα ενός DAC για μια συγκεκριμένη εφαρμογή καθορίζεται από ένα σύνολο προδιαγραφών καθώς επίσης από την ταχύτητα και το κόστος που θέτει η εφαρμογή.

3.2 DAC με Δικτύωμα R-2R

Ένα κύκλωμα που λαμβάνει μία ψηφιακή είσοδο και τη μετατρέπει σε αναλογική τάση ή ρεύμα, ονομάζεται μετατροπέας ψηφιακού σήματος σε αναλογικό.

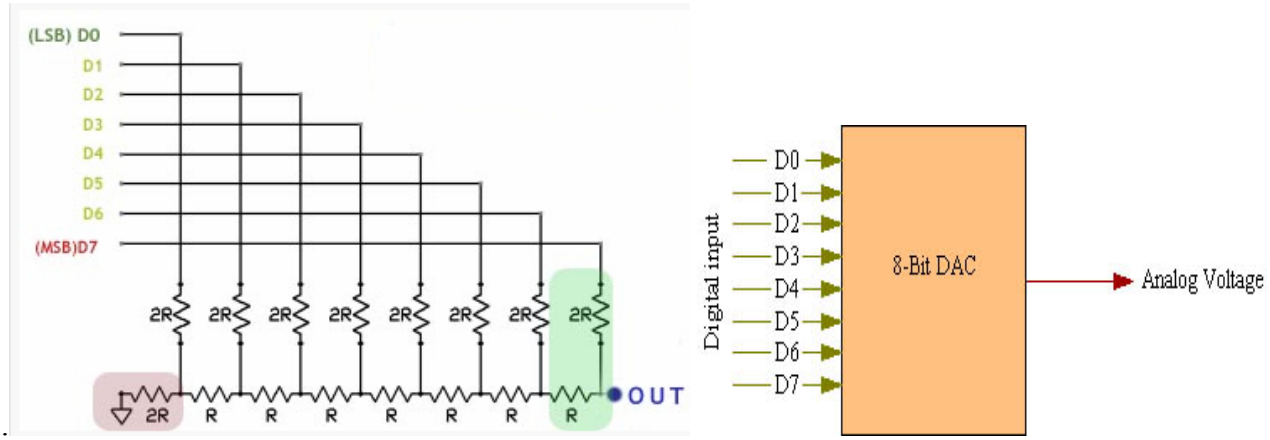
Ένα τέτοιο κύκλωμα χρησιμοποιήθηκε στην παρούσα εφαρμογή για να μετατρέπει τις τιμές που εξάγονται στην PORTD, που αποτελεί θύρα εξόδου για τις τιμές του σήματος ημιτόνου και του πριονωτού σήματος.

Σε ένα τέτοιο σύστημα κάθε ψηφιακή είσοδος αντιστοιχεί σε μια ορισμένη αναλογική έξοδο. Είναι ένα κύκλωμα το οποίο παίρνει σαν ψηφιακή είσοδο 8 bit δηλαδή αριθμούς από 0 (00000000) ως 255 (11111111) και σαν έξοδο παράγει μια τάση από 0-5V .

Η μαθηματική εξήγηση είναι αρκετά απλή: αν θέλουμε έναν 8 bit DAC να δίνει έξοδο 5V σε 255 βήματα εκτέλεσης κάθε βήμα θα είναι 5/255=0.019V. Έτσι η τάση εξόδου ενός τέτοιου DAC θα είναι ίση με τον πολλαπλασιασμό της δυαδικής εισόδου επί το βήμα. Παράδειγμα για είσοδο τον αριθμό 129 (1000 0001) η τάση εξόδου θα ήταν 129X0.019=2.451V

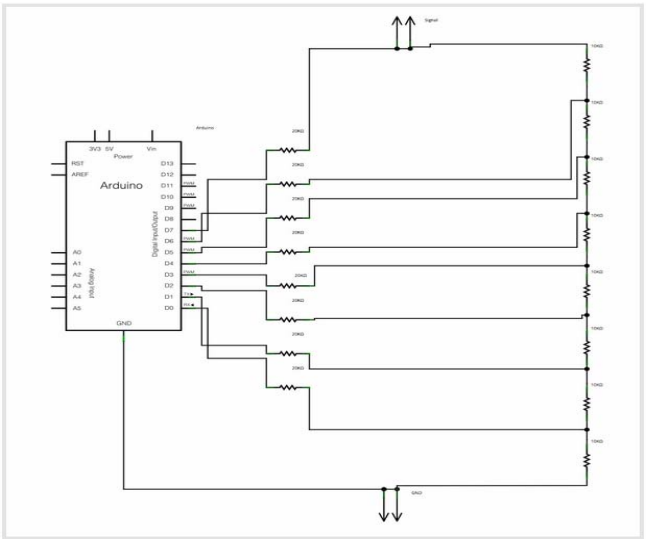
Στο κύκλωμα το οποίο υλοποιήσαμε έγινε κατασκευή και χρήση ενός **DAC R-2R 8 bit.**

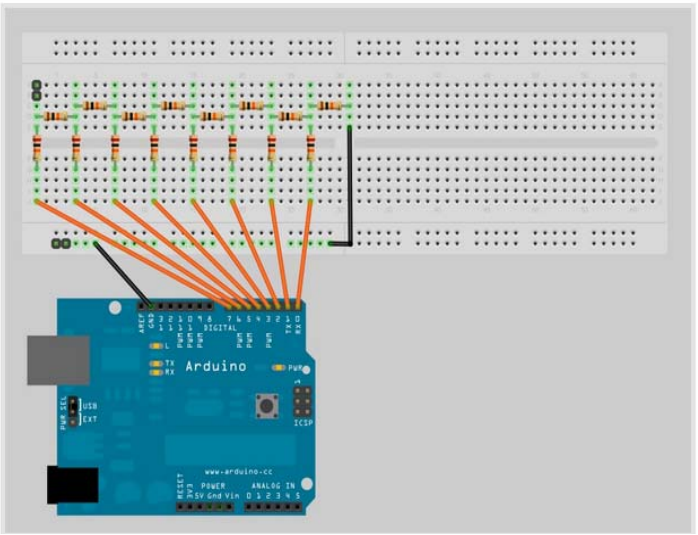
Το κύκλωμα αυτό είναι ένας 8 bit μετατροπέας ψηφιακού σε αναλογικό σήμα. Κάθε ένα από τα οκτώ bits συμβάλει στην προκύπτουσα τάση εξόδου. Αν όλα και τα 8 bits είναι σε κατάσταση HIGH τότε η έξοδος είναι περίπου ίση με την τάση αναφοράς. Αν αλλάξουμε τα ποιο σημαντικά bits σε κατάσταση LOW τότε θα έχουμε περίπου το μισό αυτής της τάσης. Πιο συγκεκριμένα για έναν DAC 8bit αν όλα και τα 8 bit είναι HIGH τότε παίρνουμε 255/256 τάση αναφοράς. Αν γίνει ενεργό το πιο σημαντικό ψηφίο σε κατάσταση LOW μας δίνει 127/256 των τάσεων αναφοράς. Γενικά για κάθε bit τιμής X μεταξύ 0-255 ο DAC θα δώσει X/255 τάση αναφοράς



3.3 Το Κύκλωμα R-2R

Τα ψηφιακά δεδομένα εισέρχονται μέσο των 8 γραμμών εισόδου (D_0-D_7) και προσχωρούν για την μετατροπή στην ισοδύναμη αναλογική τάση (Vout) στο δικτύωμα R-2R network.Οι περισσότεροι DAC είναι βασισμένοι σε αυτή τη φιλοσοφία. Το κύκλωμα R-2R είναι φτιαγμένο από ένα σύνολο αντιστάσεων οι οποίες έχουν 2 τιμές. Η μια είναι διπλασία της άλλης .Στο κύκλωμα που υλοποιήσαμε έγινε χρήση αντιστάσεων 10KΩ και 20KΩ. Παρακάτω παρουσιάζεται το κύκλωμα υλοποιημένο σε Raster και η σύνδεση του με την πλακέτα Arduino.





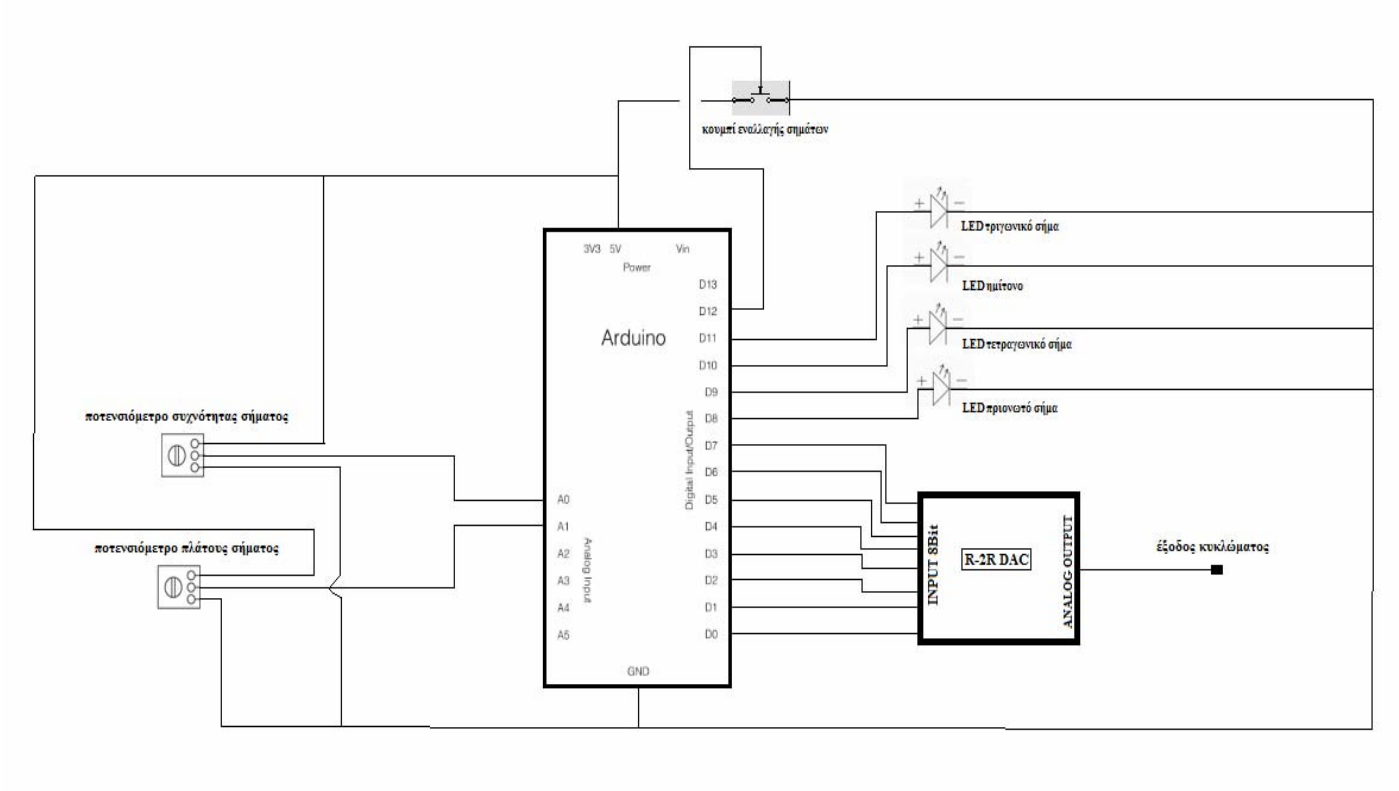
ΚΕΦΑΛΑΙΟ 4ο

4.1 Υλικά Κυκλώματος

Τεμάχια	Είδος	Λειτουργία
4	LED	Ένδειξη επιλεγμένου σήματος
8	Αντίσταση 5ΚΩ (5% ανοχή)	Κύκλωμα R-2R
8	Αντίσταση 10ΚΩ (1% ανοχή)	Κύκλωμα R-2R
2	Ποτενσιόμετρο	Ρύθμιση πλάτους-συχνότητας
1	Button	Επιλογή σήματος
6	Αντίσταση 10ΚΩ (1% ανοχή)	Περιφερικές λειτουργίες
20	Καλώδιο	Εσωτερικές συνδέσεις
1	Raster	Υλοποίηση κυκλώματος
1	Arduino Uno	Πλακέτα υλοποίησης
1	Παλμογράφος	Προβολή κυματομορφών

4.2 Περιγραφή Κυκλώματος

Από την στιγμή που το κύκλωμα παίρνει ρεύμα, αρχικοποιούνται όλες οι μεταβλητές και εκτελείται συνεχώς η συνάρτηση loop. Αρχικά όλα τα led είναι σβηστά και δεν υπάρχει κανένα σήμα στην έξοδο. Με το πάτημα του κουμπιού εναλλαγής σήματος ανάβει το πρώτο λαμπάκι το οποίο δείχνει πιο σήμα είναι στην έξοδο και οι ακροδέκτες D0-D7 δίνουν το κατάλληλο σήμα το οποίο όταν περνάει από τον μετατροπέα από ψηφιακό σε αναλογικό, μας δίνει έξοδο ένα τριγωνικό σήμα. Από τα 2 ποτενσιόμετρα (ποτενσιόμετρο ρύθμισης συχνότητας και ποτενσιόμετρο ρύθμισης πλάτους) μπορούμε να ρυθμίσουμε την συχνότητα και το πλάτος του σήματος που υπάρχει στην έξοδο. Αν ξαναπατηθεί το κουμπί εναλλαγής σήματος, ανάβει το led το οποίο μας δείχνει ότι είναι ενεργό το επόμενο σήμα- ημιτονικό σήμα. Οι ακροδέκτες D0-D7, δίνουν σήμα το οποίο όταν θα περάσει από τον μετατροπέα σήματος ψηφιακό σε αναλογικό θα μας δώσει στην έξοδο ένα ημιτονικό σήμα. Αυτό συμβαίνει και για τα άλλα δύο σήματα (τετραγωνικό και πριονωτό σήμα) με τη μόνη διαφορά την έξοδο και το αντίστοιχο led που ανάβει κάθε φορά.


κουμπί εναλλαγής σημάτων
LED τριγωνικό σήμα
LED ημίτονο
LED τετραγωνικό σήμα
LED πριονωτό σήμα
ποτενσιόμετρο συχνότητας σήματος
ποτενσιόμετρο πλάτους σήματος
Arduino
3V3 5V Vin
Power
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
INPUT 8Bit
R-2R DAC
ANALOG OUTPUT
έξοδος κυκλώματος

4.3 Κώδικας γεννήτριας σημάτων

```
int sine[254]; //Πίνακας τιμών ημιτόνου.
int outputArray[8]= { 0,  1,  2,  3,  4,  5,  6,  7 }; //Πίνακας για τα Pin εξόδου (D0-D7)
int ledArray[4] = { 8,  9,  10,  11 };  //Led-Πίνακας για τα Pin εξόδου των LED ένδειξης σημάτων
int Signal=0;  //Μεταβλητή επιλογής σήματος.


int buttonPin = 12 ; //Pin για το κουμπί εναλλαγής.
int buttonState = 0; //Μεταβλητή κατάστασης κουμπιού.



int PotFunc = 0; //Pin ποτενσιόμετρου για ρύθμιση συχνότητας σήματος.
int val = 0; //Μεταβλητή για την τιμή του ποτενσιόμετρου συχνότητας.
int PotGain =1; //Pin ποτενσιόμετρου για ρύθμιση πλάτους σήματος.
int val1 = 0;    //Μεταβλητή για την τιμή του ποτενσιόμετρου πλάτους.

//Συνάρτηση ρύθμισης πλακέτας.
void setup()
{
   //Ορισμός των pin 0 εώς 7 ως έξοδοι (ψηφιακό σήμα).
   for ( int i=0;i<8 ; i++ )
     pinMode(outputArray[i], OUTPUT);

   //Ορισμός των pin 8 εώς 11 ως έξοδοι(Led ένδειξης).
   for ( int i=0;i<4 ; i++ )
    pinMode(ledArray[i], OUTPUT);

   //Ορισμός του pin 12 ως είσοδος(κουμπί).
   pinMode(buttonPin, INPUT);

    float x;  //βοηθητική μεταβλητή.
    float y;  //βοηθητική μεταβλητή.

   //Συμπλήρωση του πίνακα sine με 256 τιμές  του ημιτόνου από 0 εώς 2(χωρίς αρνητικές τιμές).
    for(int i=0;i<255;i++)
    {
        x=float(i);
        y=sin((x/255)*2*PI);
        sine[i]=int(y*127)+127;
    }
}
//Συνάρτηση που ελέγχει την κατάσταση του κουμπιού εναλλαγής.
int statecheck() {

//Αν η μεταβλητή Signal είναι μεγαλύτερη του 4(4 σήματα) τότε την μηδενίζει.
  if (Signal>4)
    Signal=0;
/* Ανάθεση τιμής από την συνάρτηση που διαβάζει την κατάσταση του κουμπιού στην μεταβλητή
```

```
button state */
  buttonState = digitalRead(buttonPin);
  //Αν το κουμπί είναι πατημένο(κατάσταση LOW)
  if (buttonState == LOW)  {
   //Αυξάνει την μεταβλητή ώστε να πάει στο επόμενο σήμα.
   Signal++;
   //χρόνος αναμονής για πάτημα του κουμπιού.
   delay(300);
  }
  return Signal;
}

//Συνάρτηση η οποία παίρνει την τιμή ενός LED και το ανάβει αφού σβήσει όλα τα υπόλοιπα.
void ActiveLed(int Led) {
 Led--;
//Σβήνει όλα τα LED.
 for (int i=0;i<4;i++)
   digitalWrite(ledArray[i],LOW);
 if (Led < 4 )
   digitalWrite(ledArray[Led],HIGH); //Ανάβει μόνο το led που ορίσαμε στην συνάρτηση
}

//Συνάρτηση η οποία τρέχει το κυρίως πρόγραμμα συνεχώς.
void loop()
{

   /*Αρχικά κανένα σήμα δεν λειτουργεί και κανένα από τα led δεν είναι αναμμένο.
  Ελέγχεται συνεχώς η κατάσταση του κουμπιού με την statecheck() */
   Signal=statecheck();
   val1=analogRead(PotGain);  /*Ανάγνωση τιμής του ποτενσιόμετρου πλάτους και απόδοση τιμής
στην μεταβλητή val1. */
   val1=map(val1,0,1023,1,5); //αντιστοίχιση τιμών από εύρος 0-1023 στο 1-5.
   val = analogRead(PotFunc);  //συνεχής έλεγχος ποτενσιόμετρου συχνότητας.
   val = map(val, 0, 1023, 0, 20);  //αντιστοίχιση τιμών ποτενσιόμετρο στο εύρος 0 ώς 20.

 //Σε περίπτωση που πατηθεί το κουμπί αυξάνεται η τιμή του Signal και μπαίνει στην case
   switch (Signal) {
     case 1: /*Στην περίπτωση που η μεταβλητή Signal πάρει την τιμή 1 τότε τρέχει ο παρακάτω
κώδικας που δίνει έξοδο το τριγωνικό σήμα. */
      ActiveLed(Signal);   //ελέγχεται συνεχώς αν πατήθηκε το κουμπί ώστε να ανάψει άλλο LED.
      for (int i=0;i<255;i++) { //βρόγχος ανόδου.
          delayMicroseconds (val);  //η συχνότητα ορίζεται από την καθυστέρηση.
          Signal=statecheck(); //ελέγχεται συνεχώς αν πατήθηκε το κουμπί ώστε να αλλάξει σήμα.
          if(Signal!=1) break; //αν η τιμή της Signal αλλάξει τότε το πρόγραμμα βγαίνει από το Loop.
```

```
        PORTD=i/val1;  /*έξοδος τιμών στον καταχωρητή PORTD.Διαιρείται με την τιμή του ποτενσιόμετρου έτσι ώστε να μικραίνει η να μεγαλώνει αντίστοιχα το πλάτος του σήματος μου */
      }
      for (int i=255;i>0;i--) { //βρόγχος καθόδου.
        val = analogRead(PotFunc);  //συνεχής έλεγχος ποτενσιόμετρου συχνότητας.
        val = map(val, 0, 1023, 0, 20); //αντιστοίχιση τιμών ποτενσιόμετρο στο εύρος 0 εώς 20.
        delayMicroseconds (val); //η συχνότητα ορίζεται από την καθυστέρηση.
        Signal=statecheck();//ελέγχεται συνεχώς αν πατήθηκε το κουμπί ώστε να αλλάξει σήμα.
        if(Signal!=1) break; //αν η τιμή της Signal αλλάξει τότε το πρόγραμμα βγαίνει από το Loop.
        PORTD=i/val1; /*έξοδος τιμών στον καταχωρητή PORTD.Διαιρείται με την τιμή του ποτενσιόμετρου έτσι ώστε να μικραίνει η να μεγαλώνει αντίστοιχα το πλάτος του σήματος μου */

      }

      break;
     case 2: /* Στην περίπτωση που η μεταβλητή Signal πάρει την τιμή 2 τότε εκτελείται ο παρακάτω κώδικας που δίνει έξοδο το ημιτονικό σήμα. */

      ActiveLed(Signal);
      for (int i=0;i<254;i++) {
        delayMicroseconds (val);
        Signal=statecheck();
        if(Signal!=2) break;
        PORTD=sine[i]/val1; /* απόδοση τιμών του πίνακα τιμών ημιτόνου στον καταχωρητή PORTD  */
      }
       break;
     case 3: /* Τετραγωνικό σήμα. Ο βρόγχος είναι ίδιος με τους προηγούμενους με την διαφορά ότι εδώ η έξοδος είναι 0 για μία περίοδο και μέγιστη (5V) για την επόμενη και αυτό επαναλαμβάνεται μέχρι να πατηθεί το κουμπί  αλλαγής σήματος.*/
        ActiveLed(Signal);
        for (int i=0;i<255;i++) {
          if(i<128) {
            Signal=statecheck();
            if(Signal!=3) break;
            delayMicroseconds (val);
            PORTD=255/val1;
          }
          else
          {
            Signal=statecheck();
            if(Signal!=3) break;
            delayMicroseconds (val);
            PORTD=0;
          }
      }
     break;
     case 4: /*Πριονωτό σήμα. Η λογική επαναλαμβάνεται ακόμη μία φορά με την διαφορά ότι το πριονωτό σήμα αυξάνει μέχρι μέγιστη τάση και μετά πέφτει απότομα στο 0. */
       ActiveLed(Signal);
       for (int i=0;i<255;i++) {
          if(Signal!=4) break;
```

```
            Signal=statecheck();
            delayMicroseconds (val);
            PORTD=i/val1;
          }
        break;
        default: /*Κατάσταση στην οποία η μεταβλητή Signal έχει την τιμή 0.Κανένα Led δεν ανάβει και η έξοδος είναι 0*/
          ActiveLed(Signal);
          PORTD=0;
        break;
    }

}
```

4.4 Λογική του προγράμματος

Το πρόγραμμα αποτελείται από τρία μέρη και είναι τα εξής :

1. Δήλωση των global μεταβλητών.
2. Συνάρτηση setup στην οποία, καθορίζονται οι αρχικές ρυθμίσεις του προγράμματος που έχουν να κάνουν με την πλακέτα και η απόδοση τιμών, στις μεταβλητές(παράδειγμα απόδοση τιμών στον πίνακα sine).
3. Συνάρτηση loop στην οποία, εκτελείται συνεχώς για όσο λειτουργεί το Arduino το κυρίως πρόγραμμα.

4.4.1 Οι μεταβλητές που χρησιμοποιήθηκαν:

- sine[254] : Πίνακας 255 θέσεων ο οποίος παίρνει τιμές ημιτόνου από 0 εώς 255. Χρησιμεύει για να δώσουμε έξοδο τιμών ημιτόνου στον καταχωρητή PORT.
- OutputArray[8] : Πίνακας με 8 θέσεις που θα οριστούν ως έξοδοι. Αυτές οι 8 έξοδοι χρησιμοποιούνται για να δίνουν τα 4 σήματα της γεννήτριας. Pins D0-D7
- ledArray[4] : Πίνακας 4 θέσεων που ορίζονται ως έξοδοι. Συνδέονται με 4 Led τα οποία χρησιμεύουν για να δείχνουν πιο σήμα υπάρχει στην έξοδο κάθε φορά. Pins D8-D11
- Signal : Μεταβλητή η οποία παίρνει τιμές από 0-4 και καθορίζει ποια περίπτωση της case θα τρέξει κάθε φορά.
- buttonPin : Μεταβλητή η οποία παίρνει μία τιμή η οποία είναι η θέση του pin στο οποίο είναι συνδεδεμένο το κουμπί για αλλαγή σήματος. Pin D12
- buttonState : Μεταβλητή εισόδου, η οποία παίρνει την κατάσταση του κουμπιού(LOW – πατημένο, HIGH – μη πατημένο).

- PotFunc : Μεταβλητή που παίρνει μία τιμή(0 στην περίπτωση μας) η οποία είναι η τιμή εισόδου του ποτενσιόμετρου. Pin - A0 (Αναλογική είσοδος).
- val(value): Μεταβλητή η οποία παίρνει την τιμή της κατάστασης του ποτενσιόμετρου.
- PotGain : Μεταβλητή που παίρνει μία τιμή(1 στην περίπτωση μας) η οποία είναι η τιμή εισόδου του ποτενσιόμετρου. Pin – A1 (Αναλογική είσοδος).
- val1(value) : Μεταβλητή η οποία παίρνει την τιμή της κατάστασης του ποτενσιόμετρου.

4.4.2 Συνάρτηση Setup

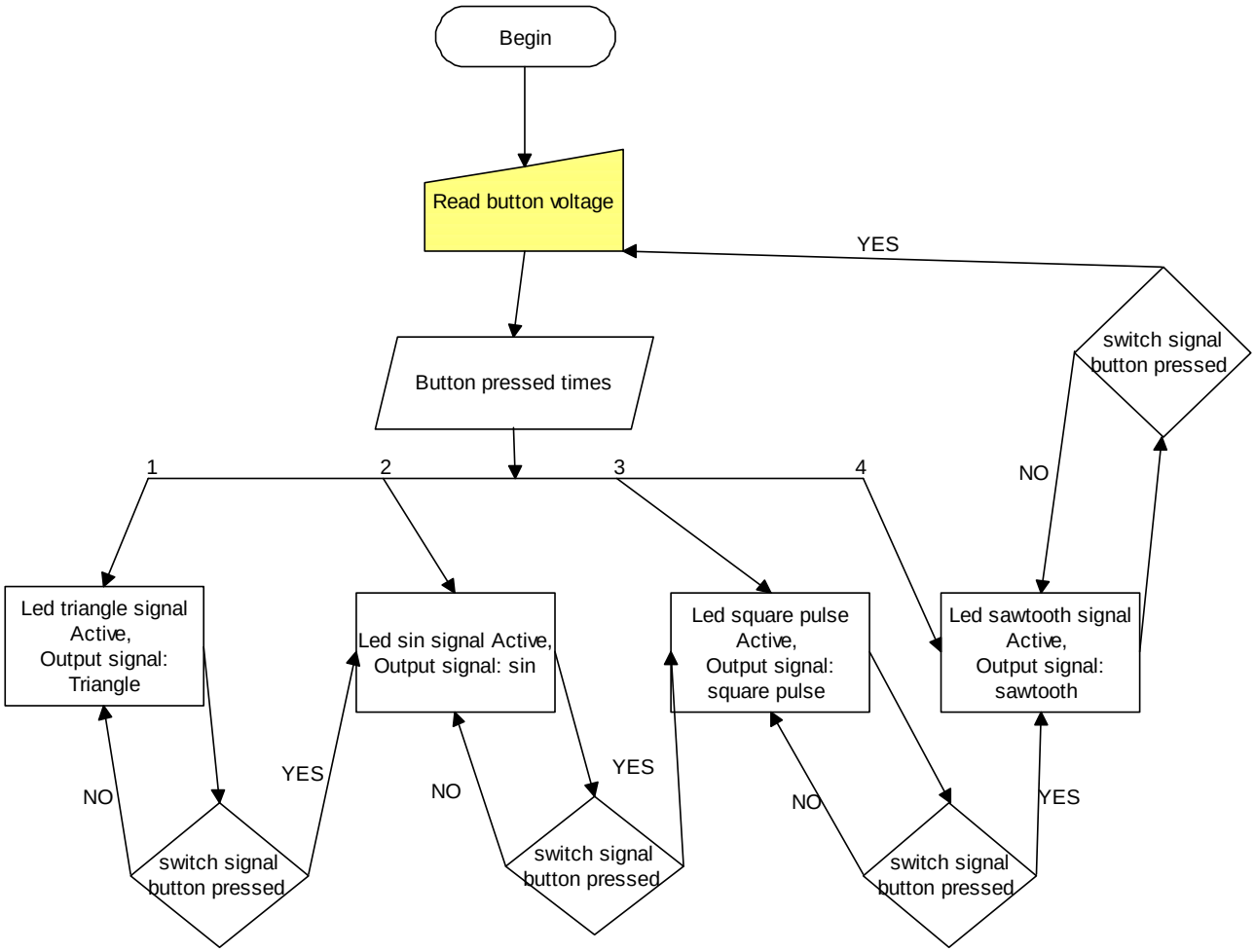
Στην συνάρτηση αυτή ορίζονται οι λειτουργίες των Pin της πλακέτας του Arduino και γίνονται αρχικοποιήσεις των μεταβλητών. Συγκεκριμένα τα ψηφιακά Pin από 0 έως 11 ορίζονται ως έξοδοι. Τα ψηφιακά Pin 0–7 έχουν να κάνουν με την έξοδο του κυκλώματος και μας δίνουν κάθε φορά μία αλληλουχία από bits που παίρνουν καταστάσεις τέτοιες ώστε να εμφανίζεται στην έξοδο το ανάλογο σήμα (τριγωνικό, ημιτονικό, τετραγωνικό, πριονωτό). Τα Pin 8-11 δίνουν έξοδο για τα 4 led τα οποία χρησιμοποιούνται ως δείκτες σηματοδότησης για το πιο σήμα υπάρχει στην έξοδο κάθε φορά.
Το ψηφιακό Pin 12 ορίζεται ως είσοδος και παίρνει τιμή από το κουμπί εναλλαγής σημάτων. Όταν το κουμπί είναι πατημένο τότε υπάρχει στο Pin τάση 5V άρα είναι σε κατάσταση High και όταν δεν είναι πατημένο τότε δεν υπάρχει τάση και άρα είναι σε κατάσταση low.
Τα αναλογικά Pin 0 και 1 ορίζονται ως είσοδοι που παίρνουν τιμές από την εσωτερική αντίσταση των 2 ποτενσιόμετρων που ρυθμίζουν την ένταση και την συχνότητα των σημάτων κάθε φορά.

4.4.3 Συνάρτηση Loop

Αφού ορισθούν οι global μεταβλητές και εκτελεστεί η συνάρτηση setup,από την στιγμή που ξεκινάει να εκτελείται η συνάρτηση loop «τρέχει» συνεχώς, σε έναν αέναο βρόχο για όσο έχει ρεύμα το arduino ή μέχρι να γίνει επανεκκίνηση. Ο κώδικας της συνάρτησης αυτής είναι και το κυρίως πρόγραμμα. Αρχικά η loop εκτελείται και ελέγχεται, από μία case, η συνάρτηση signal η οποία παίρνει τιμή από το κουμπί εναλλαγής σημάτων. Αρχικά η signal είναι 0 και έτσι δεν συμβαίνει τίποτα (default κατάσταση της case). Αν πατηθεί μία φορά το κουμπί, τότε μπαίνει στην πρώτη περίπτωση και αρχίζει να εκτελείται ο κώδικας, ο οποίος δίνει στην έξοδο ένα τριγωνικό σήμα και ανάβει το κατάλληλο led (συνάρτηση active_led) ένδειξης σήματος. Παράλληλα ελέγχονται τα 2 ποτενσιόμετρα για αλλαγές συχνότητας και πλάτους και με την συνάρτηση map προσαρμόζουμε τις τιμές που παίρνουμε από τα ποτενσιόμετρα στο εύρος που μας ενδιαφέρει να δουλέψουμε. Επίσης με την συνάρτηση delay ορίζουμε καθυστέρηση στον βρόχο έτσι ώστε να μπορούμε να αλλάζουμε συχνότητα στα σήματα. Παράλληλα με όλα αυτά, ελέγχεται το κουμπί

εναλλαγής σήματος έτσι ώστε αν πατηθεί να τρέξει η επόμενη περίπτωση, στην οποία το μόνο που αλλάζει είναι ο αλγόριθμος ο οποίος δίνει σήμα εξόδου και το led ένδειξης σήματος. Το ίδιο συμβαίνει σε κάθε πάτημα του κουμπιού μέχρι να τελειώσουν τα σήματα και το πρόγραμμα να έρθει στην αρχική κατάσταση(default).

4.5 Διάγραμμα ροής προγράμματος



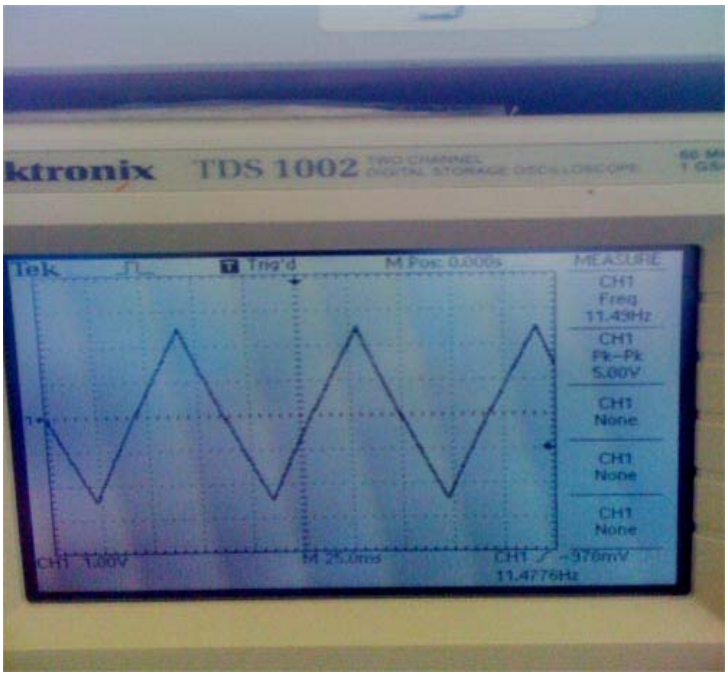
4.6 Αποτελέσματα πειράματος

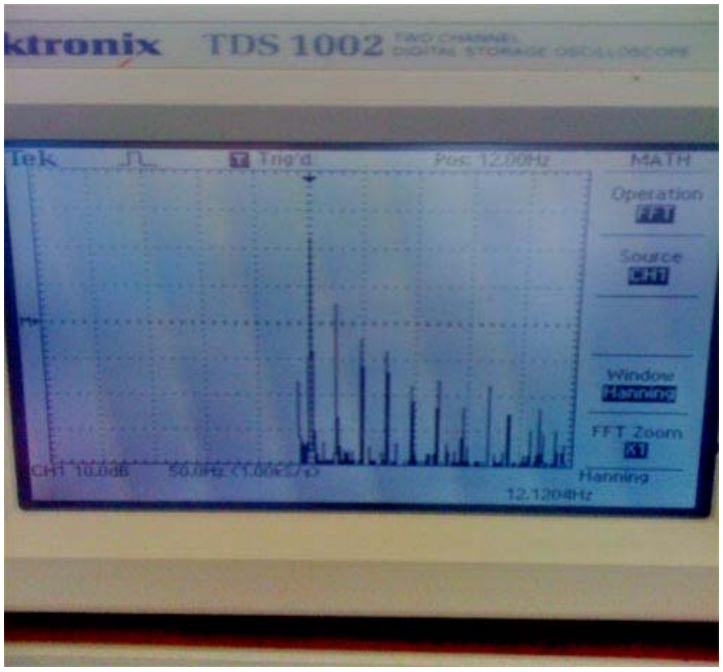
Η εκτέλεση του πειράματος πραγματοποιήθηκε στα εργαστήρια επικοινωνιών του τμήματος με την χρήση παλμογράφου. Τα αποτελέσματα συλλέχτηκαν για την κάθε κυματομορφή ξεχωριστά και σε δυο φάσεις. Η πρώτη φάση με μια πρώτη ελάχιστη συχνότητα και πλάτος και η δεύτερη για μια μέγιστη συχνότητα και πλάτος για την κάθε κυματομορφή ξεχωριστά. Τα αποτελέσματα συλλέχτηκαν στο πεδίο του χρόνου αλλά και στο πεδίο της συχνότητας. Το εύρος συχνοτήτων καθορίζεται κάθε φορά κατά περίπτωση ανάλογα την χρήση της γεννήτριας και τον σκοπό .

4.6.1 Τριγωνικό σήμα

Πεδίο χρόνου	Πεδίο συχνότητας
F= 11 Hz	F_0= 11 Hz
Vp-p= 5V	3 $F_{0=}$ 33 Hz
Vp= 2.5V	
Vrms= 0.707 Vp	5 $F_{0=}$ 55 Hz
	7 $F_{0=}$ 77 Hz
	9 $F_{0=}$ 99 Hz

F : Μικρότερη συχνότητα που πετυχαίνεται και μετριέται.
F_0 : Θεμελιώδης συχνότητα τριγωνικού σήματος.
F(3,5,7,9) : Αρμονικές που εμφανίζει το τριγωνικό σήμα.
Vp-p : Μέγιστο πλάτος Σήματος.
Vrms : Ενεργός τιμή σήματος.

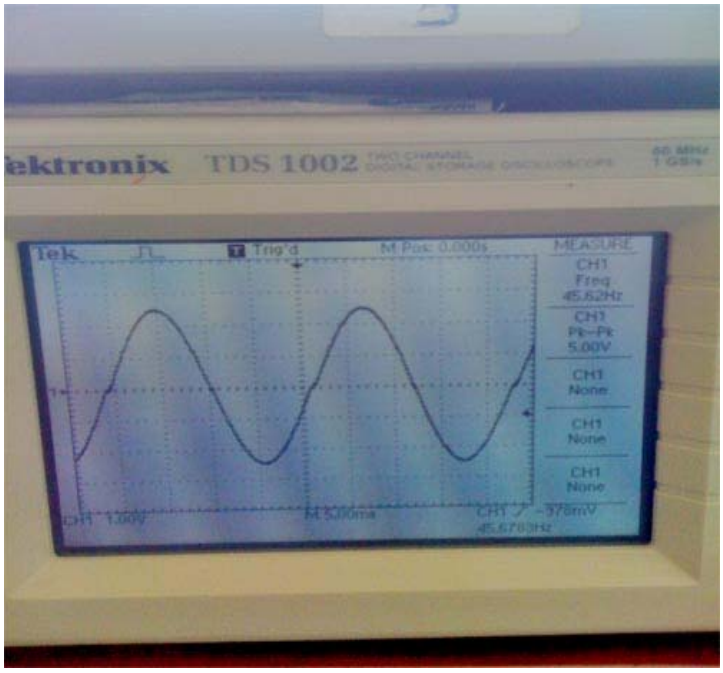


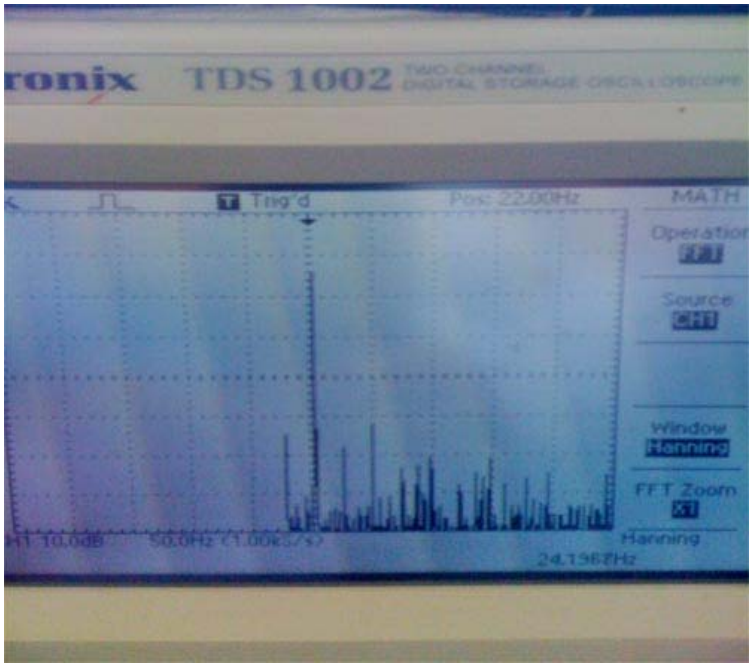


4.6.2 Ημιτονικό σήμα

Πεδίο χρόνου	Πεδίο συχνότητας
F= 24 Hz Vp-p= 5V Vp= 2.5V Vrms= 0.707 Vp	F_0= 24 Hz

F : Συχνότητα που μετράμε.
F_0 : Θεμελιώδης και μοναδική συχνότητα για το σήμα του ημιτόνου.
Vp-p : Μέγιστο πλάτος Σήματος.
Vrms : Ενεργός τιμή σήματος.

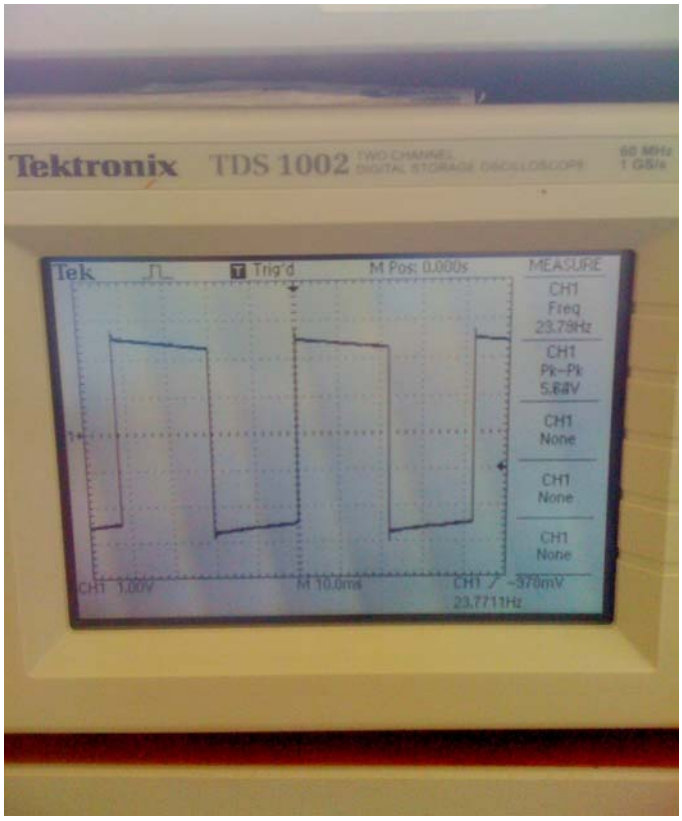


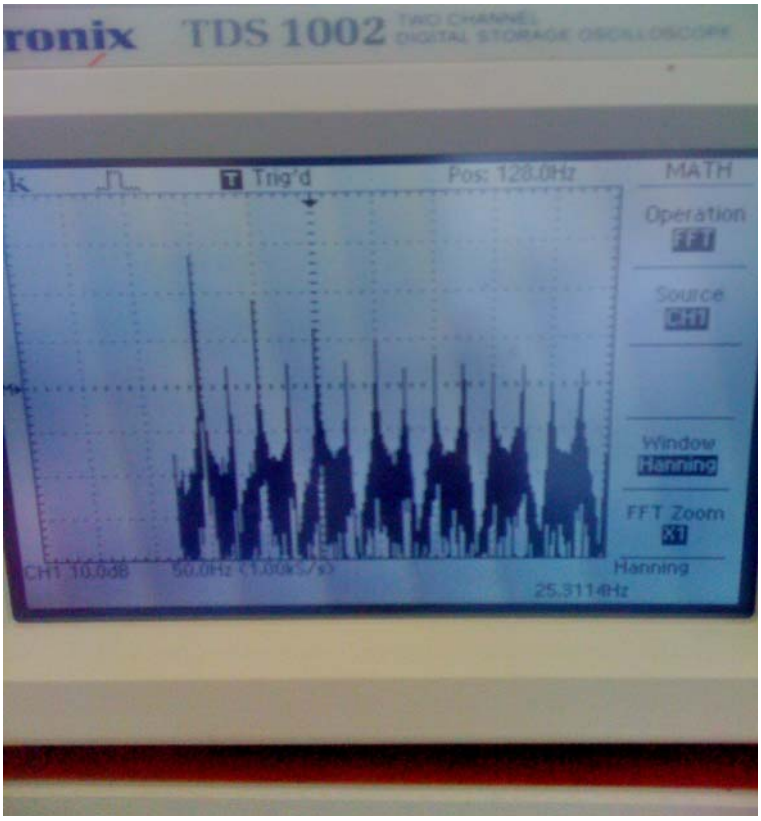


4.6.3 Τετραγωνικό σήμα

Πεδίο χρόνου	Πεδίο συχνότητας
F= 25 Hz	F_0= 25 Hz
Vp-p= 5V	3 $F_{0=}$ 75 Hz
Vp= 2.5	
Vrms= 0.707 Vp	5 $F_{0=}$ 125 Hz
	7 $F_{0=}$ 175 Hz
	9 $F_{0=}$ 225 Hz

F : Συχνότητα που μετράμε.
F_0 : Θεμελιώδης συχνότητα για το τετραγωνικό σήμα.
F(3,5,7,9) : Αρμονικές που εμφανίζει το τετραγωνικό σήμα.
Vp-p : Μέγιστο πλάτος Σήματος.
Vrms : Ενεργός τιμή σήματος.

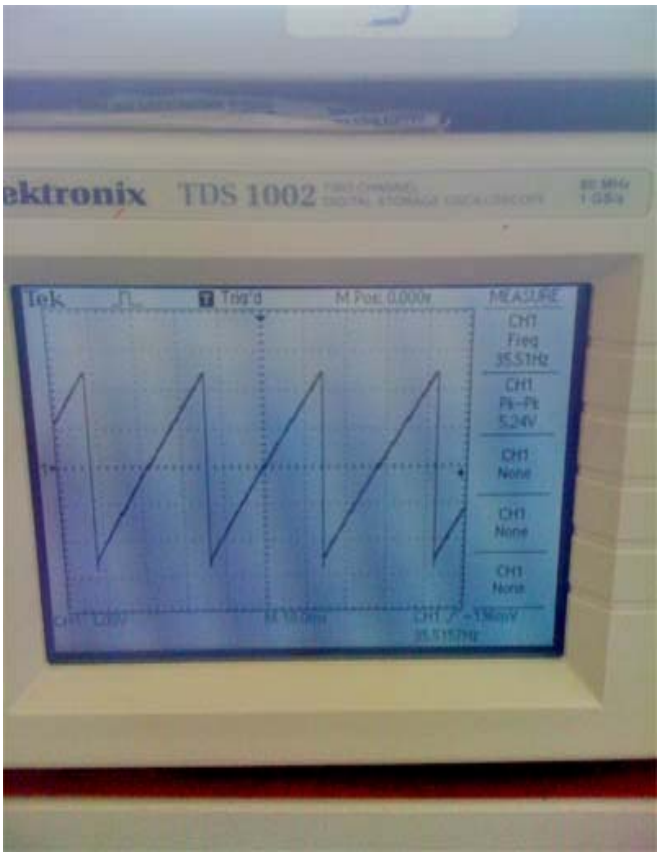


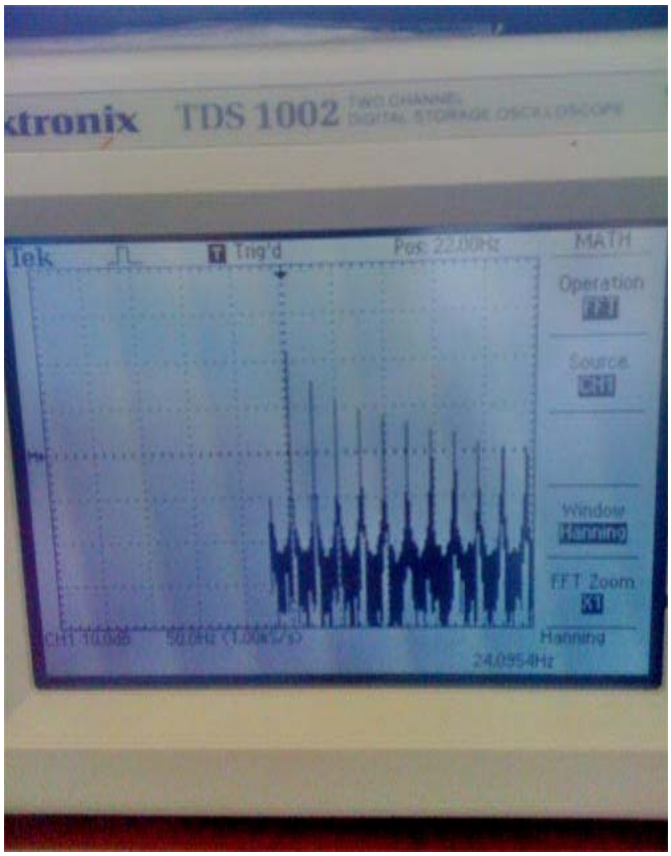


4.6.4 Πριονωτό

Πεδίο χρόνου	Πεδίο συχνότητας
F= 11 Hz	F_0= 11 Hz
Vp-p= 5V	3 $F_{0=}$ 33 Hz
Vp= 2.5V	
Vrms= 0.707 Vp	5 $F_{0=}$ 55 Hz
	7 $F_{0=}$ 77 Hz
	9 $F_{0=}$ 99 Hz

F : Συχνότητα που μετράμε.
F_0 : Θεμελιώδης συχνότητα για το πριονωτό σήμα.
F(3,5,7,9) : Αρμονικές που εμφανίζει το πριονωτό σήμα.
Vp-p : Μέγιστο πλάτος Σήματος.
Vrms : Ενεργός τιμή σήματος.





Κεφάλαιο 5ο

5.1 Σχόλια - Συμπεράσματα

Κατά την διάρκεια εκτέλεσης του πειράματος και στα 4 σήματα ξεχωριστά εκτός των δυο βασικών παραμέτρων μας πλάτος σήματος και συχνότητα έγινε μέτρηση και ορισμένων ακόμη βασικών όρων που συνοδεύουν το κάθε σήμα ξεχωριστά. Στο πεδίο του χρόνου αρχικά μετρήσαμε με την βοήθεια τις επιλογής cursors στον παλμογράφο την τάση Vp-p η οποία εκφράζει την απόσταση από την μια ελαχίστη κορυφή του σήματος ως την μέγιστη. Το επόμενο μέγεθος που υπολογίσαμε είναι το Vp το οποίο είναι το μισό της τιμής Vp-p. Με την βοήθεια υπολογισμών βρήκαμε και την ενεργό τάση RMS του σήματος. Όσο αναφορά το πεδίο τις συχνότητας τα μεγέθη τα οποία συναντήσαμε ήταν συχνότητες παράγωγες από αυτές που δίναμε κάθε φορά μέσω του ποτενσιόμετρου. Οι συχνότητες αυτές ονομάζονται αρμονικές. Πιο συγκεκριμένα τα 3 σήματα εκτός του ημίτονου έχουν παράγωγες-πολλαπλάσιες αρμονικές οι οποίες προέρχονται από την κύρια που ονομάζεται f0. Για την μέτρηση της κάθε μιας από αυτές έγινε χρήση της επιλογής cursors του παλμογράφου .
Συνηθίζεται να μετρούμε τα περιττά πολλαπλάσια των αρμονικών αυτών τα οποία είναι πολλαπλάσια της πρώτης αρμονικής που είναι και η συχνότητα του σήματος μας. Στην περίπτωση του ημιτόνου υπάρχει μόνο μια βασική αρμονική που είναι η συχνότητα του ημιτόνου .

Στο αρχικό στάδιο ασχοληθήκαμε με τον σχεδιασμό και την υλοποίηση του κυκλώματος. Οι δυσκολίες που αντιμετωπίσαμε ήταν καθαρά συμβατικού χαρακτήρα δηλαδή λειτουργικές πάνω στην πλακέτα Arduino, σε σχέση πάντα με την αρχιτεκτονική του. Ο αρχικός σχεδιασμός του μετατροπέα ψηφιακού σήματος σε αναλογικό έγινε με χρήση ολοκληρωμένου κυκλώματος. Η καθορισμένη αρχιτεκτονική και λειτουργία του Arduino δεν μας βοήθησε στη λειτουργία του κυκλώματος με ολοκληρωμένο. Στην συνέχεια έγινε μελέτη και υλοποίηση ενός μετατροπέα καθαρά με αντιστάσεις βασισμένου στο πρότυπο R-2R ώστε να έχουμε τα σωστά αποτελέσματα. Στην συνέχεια στο κομμάτι του προγραμματισμού του κυκλώματος πάρθηκαν αποφάσεις με βάσει πάντα την μετατροπή ενός σήματος από διαδοχικά bits σε μια έξοδο αναλογική.
Οι δυσκολίες που αντιμετωπίσαμε ήταν στο εύρος, στο οποίο θα είχε το κάθε σήμα. Αυτό συνέβη διότι το Arduino Uno έχει 16MHz επεξεργαστή και πρέπει να προσπελαστούν 255 τιμές για κάθε περίοδο. Ενώ τα σήματα διαφέρουν στις συχνότητες, για λόγους συμμετρίας, έγιναν έτσι, ώστε να έχουν όλα το ίδιο εύρος. Κανονικά το τετραγωνικό σήμα μπορεί να πάρει πολύ μεγάλο εύρος, λόγω του ότι έχουμε μόνο 2 καταστάσεις ενώ το πριονωτό τις μισές από το ημιτονικό και το τριγωνικό.
Ένα ακόμη πρόβλημα που προέκυψε κατά την κατασκευή του κυκλώματος μας, είναι το μέγιστο ρεύμα του κυκλώματος. Για να γίνει η μετατροπή ψηφιακού σε αναλογικό σήμα χρησιμοποιήσαμε το κύκλωμα R-2R, με τη χρήση του οποίου πολύ μεγάλο μέρος του ρεύματος μας καταναλώνεται

στις αντιστάσεις. Το πρόβλημα θα μπορούσε να λυθεί με ένα κύκλωμα ενίσχυσης η ενός ολοκληρωμένου κυκλώματος ενισχυτή, όμως δυστυχώς δεν ήταν δυνατή η ενσωμάτωση του στο raster του κυκλώματος μιας και ο χώρος που είχαμε ήταν πολύ περιορισμένος και η χρήση του κρίθηκε ότι δεν είναι μείζονος σημασίας στα πλαίσια αυτής της πτυχιακής.

Βιβλιογραφία

Ηρακλής Γ. Δημόπουλος «Σήματα, Συστήματα και Κυκλώματα συνεχούς χρόνου», Έκδοσης Unitext, Σελ 62-72

Δρ. Απόστολος Γεωργιάδης, Δημήτριος Μάνος «Επικοινωνίες Ι Εργαστηριακό βοήθημα», Σέρρες 2010, Σελ 19-32

Official page of www.atmel.com, Pdf file «Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet»

Ανδρέα Κ. Ρωμανίδη «Διάδοση ηλεκτρομαγνητικών κυμάτων» εκδόσεις Ζήτη, Θεσσαλονίκη 2006, Σελ 255-273

Simon Haykin, Michel Moher, «Συστήματα επικοινωνίας», 5η έκδοση, εκδόσεις Παπασωτηρίου, Αθήνα 2010, Σελ 10-50

Διαδίκτυο :
www.avsite.gr
www.wikipedia.com
www.arduino.cc
www.microplanet.gr/tutorials/microcontrollers/arduino
www.deltahacker.gr/2009/08/01/arduino-intro/
www.cs.uoi.gr/tech_reports//publications/MT-2012-14.pdf