



ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**ΤΙΤΛΟΣ : ΦΑΣΜΑΤΙΚΗ ΑΝΑΛΥΣΗ ΣΗΜΑΤΟΣ
ΜΕ ΜΙΚΡΟΕΛΕΓΚΤΗ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΤΩΝ ΦΟΙΤΗΤΡΙΩΝ

ΣΟΦΙΑΣ ΛΙΟΝΤΟΥ

και

ΓΕΩΡΓΙΑΣ ΠΠΕΡΙΔΟΥ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : κ. ΓΕΩΡΓΙΟΣ ΤΑΤΣΗΣ

ΑΡΤΑ, ΙΟΥΝΙΟΣ 2014

Αφιερωμένη

στους γονείς μας και στις αδερφές μας

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστούμε θερμά τον καθηγητή μας **κ. Τάτση Γεώργιο**, ο οποίος μας βοήθησε στην επιλογή του θέματος της πτυχιακής εργασίας καθώς και την συμπαράσταση και την καθοδήγηση του που μας έδειξε καθ'όλη την διάρκεια εκπόνησης της εργασίας.

Θερμές ευχαριστίες, επίσης, απευθύνουμε σε όλους τους **καθηγητές** που είχαμε όλα τα χρόνια της μέχρι στιγμής ακαδημαϊκής μας ζωής, για τις γνώσεις που μας μετέδωσαν και αποτέλεσαν πρότυπα και πηγή έμπνευσης για εμάς.

Τέλος ένα μεγάλο και εγκάρδιο ευχαριστώ αξίζουν **οι γονείς μας και οι αδερφές μας**, που μας στήριξαν ηθικά δίνοντάς μας κουράγιο να συνεχίσουμε σε όλες τις ευχάριστες στιγμές που μπορεί να μας προσφέρει η ζωή.

Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Α

ΕΙΣΑΓΩΓΗ- ΣΚΟΠΟΣ.....	7
ΚΕΦΑΛΑΙΟ 1 : ΣΗΜΑΤΑ.....	8
1.1 ΕΙΣΑΓΩΓΗ.....	8
1.2 ΤΑΞΙΝΟΜΗΣΗ ΣΗΜΑΤΩΝ.....	8
1.2.1 ΣΗΜΑΤΑ ΣΥΝΕΧΟΥΣ ΧΡΟΝΟΥ Η ΑΝΑΛΟΓΙΚΑ ΣΗΜΑΤΑ.....	9
1.2.2 ΣΗΜΑΤΑ ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ.....	9
1.2.3 ΨΗΦΙΑΚΑ ΣΗΜΑΤΑ.....	10
1.2.4 ΠΕΡΙΟΔΙΚΑ ΚΑΙ ΜΗ ΠΕΡΙΟΔΙΚΑ ΣΗΜΑΤΑ.....	10-11
1.2.5 ΔΙΑΚΡΙΤΑ ΣΗΜΑΤΑ.....	11-12
ΚΕΦΑΛΑΙΟ 2 : ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER.....	13
2.1 ΤΡΟΠΟΙ ΠΕΡΙΓΡΑΦΗΣ ΕΝΟΣ ΣΗΜΑΤΟΣ.....	13
2.2 ΑΝΑΛΥΣΗ FOURIER.....	14
2.3 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ(DTFT).....	14
2.3.1 ΕΦΑΡΜΟΓΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ FOURIER.....	14-15
2.3.2 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER ΠΕΡΙΟΔΙΚΩΝ ΣΗΜΑΤΩΝ.....	15-16
2.3.3 ΙΔΙΟΤΗΤΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ FOURIER(DTFT).....	16-17
ΚΕΦΑΛΑΙΟ 3 : ARDUINO.....	18
3.1 ΟΡΙΣΜΟΣ.....	18
3.2 ΙΣΤΟΡΙΚΟ.....	18-19
3.3 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	19-20-21-22-23
3.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ARDUINO.....	23
3.5 ΕΚΔΟΣΕΙΣ.....	23-24
3.6 ΥΛΙΚΟ.....	24-25
3.7 ΠΛΑΚΕΤΕΣ ARDUINO.....	25-26-27

3.8 SHIELDS.....	28
3.9 ARDUINO UNO.....	29
3.9.1 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO UNO.....	30
3.9.2 ΜΝΗΜΗ.....	30
3.9.3 ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ.....	30-31
3.9.4 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ARDUINO UNO.....	32-33
3.9.5 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ IDE.....	34
3.9.6 ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ARDUINO.....	34
ΚΕΦΑΛΑΙΟ 4 : MATLAB.....	35
4.1 ΟΡΙΣΜΟΣ MATLAB.....	35
4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ MATLAB.....	36
4.3 ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ.....	36-37
4.4 ΧΡΗΣΙΜΕΣ ΕΝΤΟΛΕΣ ΣΤΟ MATLAB.....	38
4.5 ΑΠΛΕΣ ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ.....	38-39
4.6 ΣΥΝΑΡΤΗΣΕΙΣ ΣΤΟ MATLAB ΚΑΙ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥΣ.....	39-40-41
4.7 ΕΙΔΙΚΕΣ ΣΤΑΘΕΡΕΣ ΚΑΙ ΜΕΤΑΒΛΗΤΕΣ.....	41
4.8 ΔΙΑΝΥΣΜΑΤΑ.....	42
4.8.1 ΔΙΑΝΥΣΜΑΤΑ ΓΡΑΜΜΗΣ.....	42
4.8.2 ΔΙΑΝΥΣΜΑΤΑ ΣΤΗΛΗΣ.....	43
4.8.3 ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ ΔΙΑΝΥΣΜΑΤΩΝ.....	43
4.8.4 ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΝΥΣΜΑΤΩΝ.....	43-44
4.9 ΓΡΑΦΗΜΑΤΑ ΣΤΟ MATLAB.....	44
4.9.1 ΣΥΝΑΡΤΗΣΕΙΣ ΓΡΑΦΗΜΑΤΩΝ.....	44-45-46-47-48-49-50
ΚΕΦΑΛΑΙΟ 5 : ΚΩΔΙΚΑΣ ARDUINO.....	51
5.1 ΕΙΣΑΓΩΓΗ.....	51

5.2 ΚΩΔΙΚΑΣ.....	51-52
ΚΕΦΑΛΑΙΟ 6 : ΚΩΔΙΚΑΣ ΜΑΤΛΑΒ.....	53
6.1 ΕΙΣΑΓΩΓΗ.....	53
6.2 ΚΩΔΙΚΑΣ.....	53-54
6.3 ΣΥΜΠΕΡΑΣΜΑ- ΑΠΟΤΕΛΕΣΜΑ.....	54
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	55

ΦΑΣΜΑΤΙΚΗ ΑΝΑΛΥΣΗ ΣΗΜΑΤΟΣ ΜΕ ΜΙΚΡΟΕΛΕΓΚΤΗ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

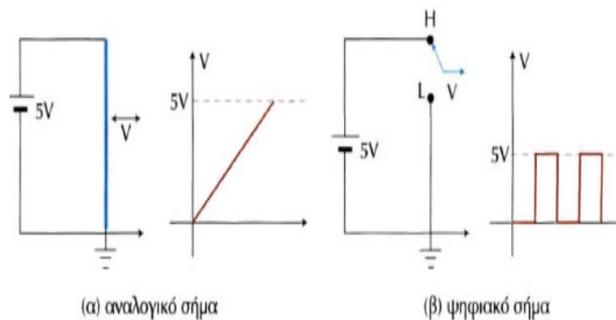
ΕΙΣΑΓΩΓΗ - ΣΚΟΠΟΣ

Σκοπός της εργασίας είναι η φασματική ανάλυση σήματος με χρήση μικροελεγκτή. Πιο συγκεκριμένα ο μικροελεγκτής πραγματοποιεί την ανάλυση στο πεδίο συχνοτήτων σε πραγματικό χρόνο και τα αποτελέσματα γίνονται διαθέσιμα στο χρήστη με τη βοήθεια ηλεκτρονικού υπολογιστή.

Για τις ανάγκες της εργασίας, θα χρησιμοποιήσουμε την πλακέτα εφαρμογών Arduino η οποία διαθέτει κατάλληλο μικροελεγκτή. Με τη βοήθεια της αναλογικής εισόδου γίνεται δειγματοληψία του αναλογικού σήματος στο οποίο εφαρμόζουμε το μετασχηματισμό Fourier διακριτών σημάτων (DFT) με τη μέθοδο FFT (Fast Fourier Transform). Η δειγματοληψία και ανάλυση γίνεται σε πραγματικό χρόνο και το φάσμα στέλνεται σε ηλεκτρονικό υπολογιστή μέσω σειριακής θύρας. Για τον μικροελεγκτή υλοποιούμε κατάλληλο πρόγραμμα σε γλώσσα C. Για την οπτικοποίηση των αποτελεσμάτων υλοποιείται αντίστοιχο πρόγραμμα σε γλώσσα MATLAB στον υπολογιστή. Ο χρήστης είναι σε θέση να βλέπει τη γραφική παράσταση του φάσματος ενός σήματος καθώς μεταβάλλεται με την πάροδο του χρόνου.

ΚΕΦΑΛΑΙΟ 1

ΣΗΜΑΤΑ



1.1 ΕΙΣΑΓΩΓΗ [1]

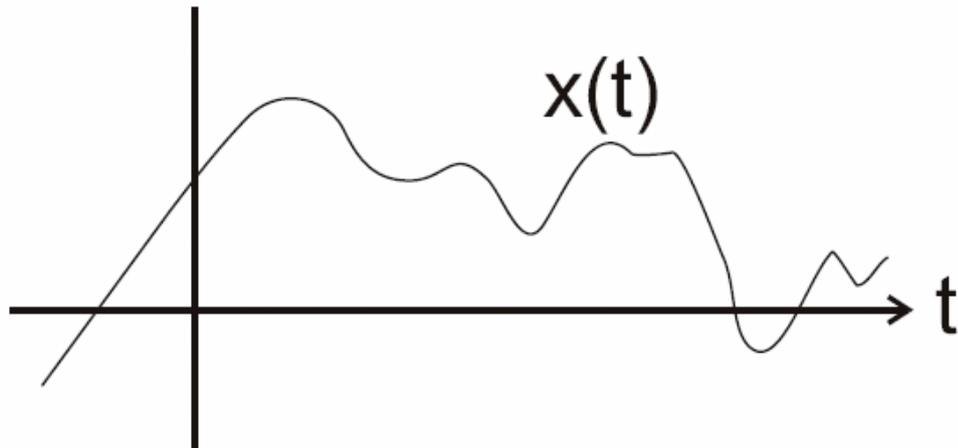
Ως σήμα ορίζεται ένα φυσικό μέγεθος το οποίο μεταβάλλεται σε σχέση με το χρόνο ή το χώρο ή, με οποιαδήποτε άλλη ανεξάρτητη μεταβλητή ή μεταβλητές. Για παράδειγμα, το σήμα ομιλίας αντιστοιχεί στις μεταβολές της ακουστικής πίεσης σε σχέση με το χρόνο και προέρχεται από τις κινήσεις των φωνητικών χορδών. Το σήμα εικόνας αντιστοιχεί στις μεταβολές της φωτεινότητας σε σχέση με τις δύο χωρικές μεταβλητές. Άλλα παραδείγματα σημάτων είναι τα σεισμικά σήματα, τα ιατρικά σήματα (όπως το καρδιογράφημα), επίσης ο ετήσιος δείκτης τιμών καταναλωτή, ο δείκτης του ποσοστού ανεργίας ανά μήνα κ.λπ. Από μαθηματική άποψη, ένα σήμα εκφράζεται ως συνάρτηση μιας ή περισσότερων ανεξάρτητων μεταβλητών. Ανάλογα με το πλήθος των ανεξαρτήτων μεταβλητών τα σήματα χαρακτηρίζονται ως μονοδιάστατα σήματα (1-D), δισδιάστατα (2-D), πολυδιάστατα σήματα.

1.2 ΤΑΞΙΝΟΜΗΣΗ ΣΗΜΑΤΩΝ [1]

Ανάλογα με τον τύπο της ανεξάρτητης ή της εξαρτημένης μεταβλητής της συνάρτησης μπορούμε να κατατάξουμε τα σήματα στις παρακάτω κατηγορίες:

1.2.1 ΣΗΜΑΤΑ ΣΥΝΕΧΟΥΣ ΧΡΟΝΟΥ Η΄ ΑΝΑΛΟΓΙΚΑ [1]

Σήματα συνεχούς χρόνου ή αναλογικά σήματα είναι τα σήματα των οποίων η ανεξάρτητη μεταβλητή μεταβάλλεται σ' ένα συνεχές διάστημα. Στα μονοδιάστατα σήματα το πεδίο ορισμού του σήματος είναι διάστημα της ευθείας των πραγματικών αριθμών. Στο σχήμα 2.1 έχει σχεδιαστεί ένα αναλογικό σήμα. Επειδή η ανεξάρτητη μεταβλητή t συνήθως είναι ο χρόνος τα σήματα αυτά ονομάζονται σήματα συνεχούς χρόνου ή σήματα συνεχούς μεταβλητής.

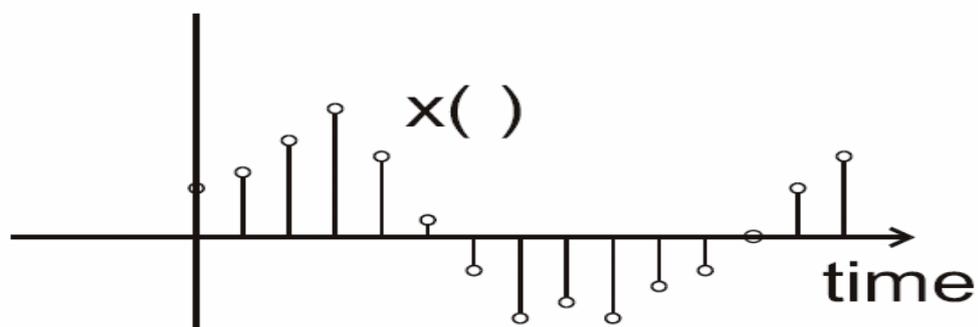


Σχήμα 2.1 Γραφική παράσταση ενός αναλογικού σήματος

1.2.2 ΣΗΜΑΤΑ ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ [1]

Σήματα διακριτού χρόνου είναι τα σήματα των οποίων το πεδίο ορισμού είναι κάποιο διακριτό σύνολο, (π.χ. το σύνολο των ακεραίων αριθμών), ενώ η εξαρτημένη μεταβλητή είναι δυνατόν να λαμβάνει οποιαδήποτε τιμή.

Το σήμα στο σχήμα 2.2 είναι ένα σήμα διακριτού χρόνου.

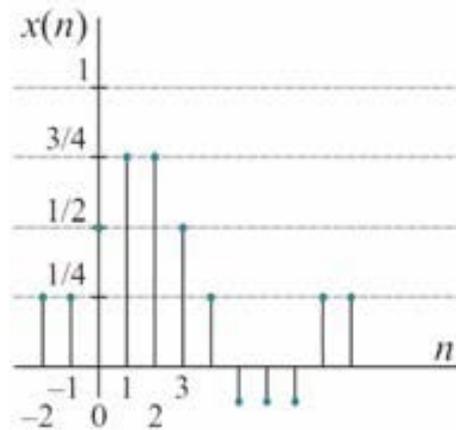


Σχήμα 2.2 Σήμα διακριτού χρόνου

1.2.3 ΨΗΦΙΑΚΑ ΣΗΜΑΤΑ [1]

Ψηφιακά σήματα είναι τα σήματα στα οποία τόσο η ανεξάρτητη μεταβλητή, όσο και η εξαρτημένη μεταβλητή μπορούν να λαμβάνουν μόνο διακριτές τιμές.

Στο σχήμα 2.3 φαίνεται ένα ψηφιακό σήμα.



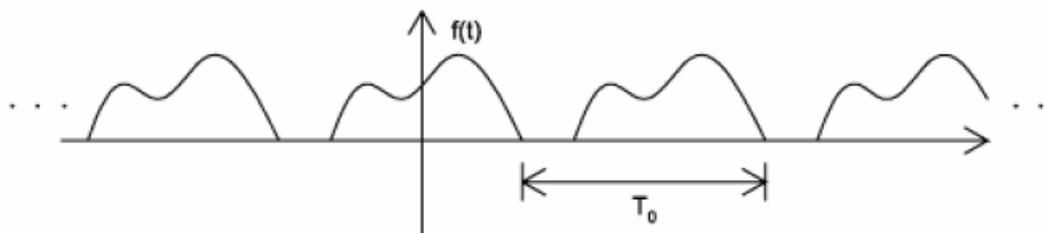
Σχήμα 2.3 Ψηφιακό σήμα

1.2.4 ΠΕΡΙΟΔΙΚΑ ΚΑΙ ΜΗ ΠΕΡΙΟΔΙΚΑ ΣΗΜΑΤΑ [1]

Ένα αναλογικό σήμα $x(t)$ λέγεται περιοδικό, όταν υπάρχει ένας θετικός αριθμός T για τον οποίο ισχύει $x(t) = x(t+T)$ για κάθε τιμή του t . Στο σχήμα 2.4 έχει σχεδιαστεί ένα περιοδικό σήμα. Ο σταθερός αριθμός T λέγεται περίοδος. Η ελάχιστη δυνατή περίοδος είναι γνωστή ως θεμελιώδης περίοδος και συμβολίζεται με T_0 .

Στην πράξη πολλές φορές αναφερόμαστε απλώς στην περίοδο και εννοούμε τη θεμελιώδη.

Περιοδικό σήμα: $f(t)=f(T+t)$



Μη περιοδικό σήμα

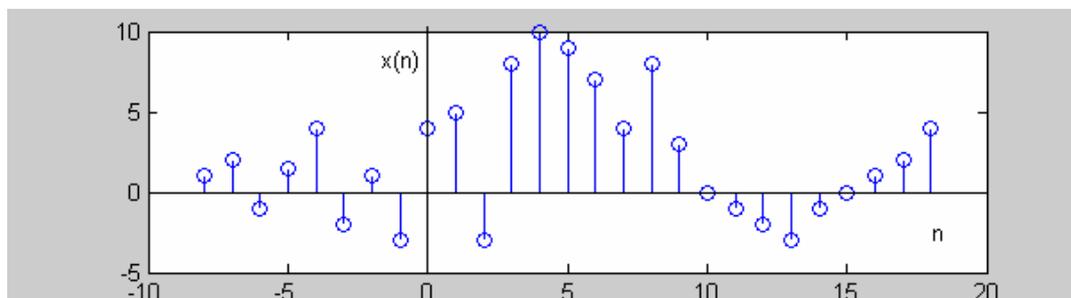


Σχήμα 2.4 Περιοδικό σήμα συνεχούς χρόνου

1.2.5 ΔΙΑΚΡΙΤΑ ΣΗΜΑΤΑ [2]

Διακριτό σήμα ή σήμα διακριτού χρόνου ονομάζουμε μία ακολουθία πραγματικών ή μιγαδικών τιμών $x(n)$, $n \in \mathbb{Z}$ και $x(n) \in \mathbb{C}$. Το διακριτό σήμα είναι δηλαδή μία συνάρτηση της οποίας η ανεξάρτητη μεταβλητή είναι ακέραιος αριθμός. Στην Ψηφιακή Επεξεργασία Σημάτων (ΨΕΣ) η ανεξάρτητη μεταβλητή n ονομάζεται και «χρόνος» αν και μπορεί να παριστάνει συντεταγμένες χώρου, αύξοντες αριθμούς κ.α. Ένα διακριτό σήμα δεν ορίζεται για τιμές του n που δεν είναι ακέραιες.

Η γραφική παράσταση ενός διακριτού σήματος έχει την μορφή που δείχνεται στο σχήμα 2.5



Σχήμα 2.5 Γραφική παράσταση διακριτού σήματος

Τα διακριτά σήματα προέρχονται από α) μεγέθη που από τη φύση τους είναι αριθμήσιμα, π.χ. έσοδα ανά ημέρα, κίνηση ανά ώρα, β) μεγέθη που μεταβάλλονται σε σχέση με μία συνεχή μεταβλητή (συνεχή σήματα) ύστερα από μία διαδικασία δειγματοληψίας. Ο πιο συνήθης τρόπος μετατροπής ενός αναλογικού σήματος σε διακριτό είναι η περιοδική ή ομοιόμορφη δειγματοληψία. Λαμβάνονται δείγματα του σήματος σε διαδοχικές τιμές της ανεξάρτητης μεταβλητής. Αν η απόσταση δύο διαδοχικών τιμών είναι σταθερή ίση με T_s και $x_a(t)$ είναι η

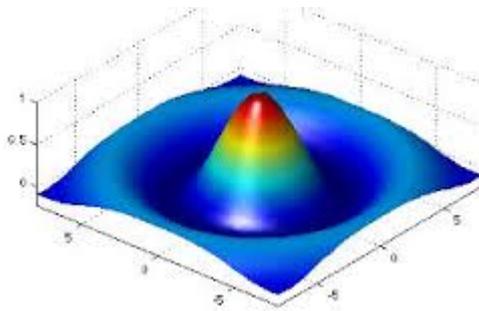
συνάρτηση της συνεχούς μεταβλητής t , το διακριτό σήμα $x(n)$ προκύπτει από τη σχέση $x(n)=x_a(nT_s)$.

Η ποσότητα $f_s = \frac{1}{T_s}$ λέγεται ρυθμός δειγματοληψίας. Αν ένα συνεχές ή διακριτό σήμα παίρνει τιμές από ένα πεπερασμένου πλήθους σύνολο τιμών, τότε λέγεται σήμα διακριτών τιμών. Μ' άλλα λόγια το πεδίο τιμών του σήματος είναι ένα σύνολο πεπερασμένου πλήθους στοιχείων.

Ένα διακριτό σήμα διακριτών τιμών λέγεται ψηφιακό. Ένα ψηφιακό σήμα πεπερασμένου μήκους μπορεί να αποθηκευτεί στη μνήμη ενός ψηφιακού υπολογιστή με ακρίβεια όση η μικρότερη απόσταση που υπάρχει μεταξύ των τιμών του. Η ψηφιοποίηση μη ψηφιακών σημάτων απαιτεί την δειγματοληψία του πεδίου ορισμού τους και τον κβαντισμό του πεδίου τιμών τους.

ΚΕΦΑΛΑΙΟ 2

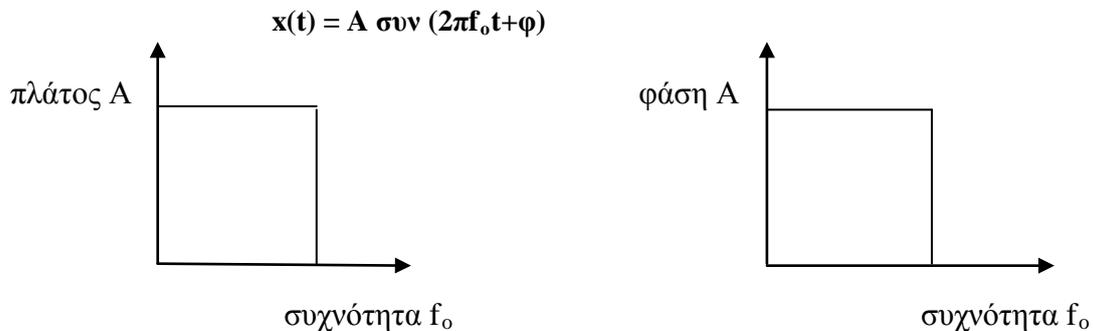
ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER



2.1 ΤΡΟΠΟΙ ΠΕΡΙΓΡΑΦΗΣ ΕΝΟΣ ΣΗΜΑΤΟΣ [4]

Υπάρχουν δύο τρόποι περιγραφής ενός αιτιοκρατικού σήματος. Ο πρώτος τρόπος περιγραφής πραγματοποιείται στο πεδίο του χρόνου, ενώ ο δεύτερος στο πεδίο της συχνότητας. Ο πρώτος τρόπος είναι άμεσα αντιληπτός και η χρονική μεταβολή του σήματος δίδεται είτε μέσω αναλυτικής σχέσης (μαθηματικός τύπος) $x(t) = A \sin(2\pi f_0 t + \pi/4)$.

Η περιγραφή των σημάτων στο πεδίο της συχνότητας περιλαμβάνει, κατά περίπτωση, τη χρήση της σειράς *Fourier* ή του μετασχηματισμού *Fourier* μέσω των οποίων ένα σήμα περιγράφεται από το φασματικό του περιεχόμενο. Η συνάρτηση η οποία περιέχει τη φασματική περιγραφή ενός σήματος ονομάζεται *φάσμα του σήματος*.



Το φάσμα του σήματος $x(t)$

2.2 ΑΝΑΛΥΣΗ FOURIER [5]

Η **ανάλυση Fourier** είναι ένα πεδίο των εφαρμοσμένων μαθηματικών το οποίο προέκυψε από την προσπάθεια αναπαράστασης μίας συνάρτησης ως άθροισματος απλούστερων περιοδικών τριγωνομετρικών συναρτήσεων. Επομένως κεντρική ιδέα στην ανάλυση Φουριέ είναι η προσπάθεια για κατανόηση των ιδιοτήτων μίας συνάρτησης (η οποία μπορεί να αναπαριστά π.χ. ένα σήμα) μέσω διάσπασής της σε γνωστά, στοιχειώδη μέρη όπως είναι η αποσύνθεση. Η αντίστροφη διαδικασία, η κατασκευή μίας συνάρτησης από γνωστές, βασικές συναρτήσεις, ονομάζεται σύνθεση. Με τον όρο ανάλυση Φουριέ αναφερόμαστε και στις δύο διεργασίες. Η μέθοδος αυτή εφαρμόστηκε για πρώτη φορά από τον Ζοζέφ Φουριέ στην προσπάθειά του να ερευνήσει τη διάδοση της θερμότητας.

2.3 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER [5]

Ο όρος **μετασχηματισμός Fourier** αναφέρεται σε μία αυστηρώς ορισμένη μαθηματική διεργασία η οποία αποσυνθέτει μία συνάρτηση σε άθροισμα απείρων περιοδικών ημιτονοειδών και συνημιτονοειδών συναρτήσεων. Το αποτέλεσμα του μετασχηματισμού είναι μία νέα συνάρτηση με διαφορετικό πεδίο ορισμού η οποία περιγράφει το κατά πόσο συμμετέχει κάθε στοιχειώδες ημίτονο στον σχηματισμό της αρχικής συνάρτησης.

Η σειρά Fourier εφαρμόζεται για περιοδική f και δίνει ως αποτέλεσμα μία νέα συνάρτηση με διακριτό πεδίο τιμών αντί για συνεχές. Αυτό σημαίνει ότι πεδίο τιμών σε μια σειρά Fourier είναι οι φυσικοί αριθμοί και όχι οι πραγματικοί.

Για συναρτήσεις διακριτής ανεξάρτητης μεταβλητής, όπου οι φυσικοί αριθμοί είναι το πεδίο ορισμού της f υπάρχουν :

- Ο μετασχηματισμός Fourier διακριτού χρόνου με συνεχές πεδίο τιμών και κατάλληλος για απεριοδικές συναρτήσεις.
- Ο διακριτός μετασχηματισμός Fourier με διακριτό πεδίο τιμών και κατάλληλος για περιοδικές συναρτήσεις.

Για κάθε μία περίπτωση υπάρχει και ο αντίστροφος μετασχηματισμός, ο οποίος δέχεται ως είσοδο το φάσμα και δίνει ως έξοδο την αρχική συνάρτηση f .

2.3.1 ΕΦΑΡΜΟΓΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ FOURIER [5]

Ο μετασχηματισμός Fourier μπορεί να χρησιμοποιηθεί στη φυσική, στη στατιστική, στην κρυπτογραφία, στην επεξεργασία εικόνας, στην επεξεργασία σήματος κ.α. διότι προσφέρει χρήσιμες ιδιότητες των ολοκληρωτικών μετασχηματισμών:

- Οι μετασχηματισμοί είναι γραμμικοί τελεστές και επίσης είναι μοναδιαίοι.
- Οι μετασχηματισμοί είναι συνήθως αντιστρέψιμοι.
- Από το θεώρημα της συνέλιξης, οι μετασχηματισμοί Fourier μετατρέπουν την συνέλιξη σε απλό πολλαπλασιασμό, το οποίο σημαίνει ότι παρέχουν ένα αποτελεσματικό τρόπο για να υπολογιστούν διαδικασίες που βασίζονται στη συνέλιξη, όπως πολλαπλασιασμός πολυωνύμων και πολλαπλασιασμός μεγάλων αριθμών.
- Οι εκθετικές συναρτήσεις είναι ιδιοσυναρτήσεις της παραγωγίσης, το οποίο σημαίνει ότι αυτή η αναπαράσταση μετασχηματίζει γραμμικές διαφορικές εξισώσεις με σταθερούς συντελεστές σε κανονικές αλγεβρικές.

2.3.2 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER ΠΕΡΙΟΔΙΚΩΝ ΣΗΜΑΤΩΝ [6]

Έστω το σήμα με μετασχηματισμό Fourier $X(j\omega) = 2\pi\delta(\omega - \omega_0)$

➤ Άρα $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (j\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} 2\pi\delta(\omega - \omega_0) e^{j\omega t} d\omega = e^{j\omega_0 t}$.

➤ Στη γενική περίπτωση

$$X(j\omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\omega - k\omega_0) \Rightarrow x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$$

➤ Άρα ο Μ.Φ. ενός περιοδικού σήματος $x(t)$ με αναπαράσταση σε σειρές Fourier $x(t) \leftrightarrow a_k$ είναι άθροισμα κρουστικών αποκρίσεων στα σημεία $k\omega_0$ με επιφάνεια $2\pi a_k$.

➤ **Περιοδικά** σήματα – Σειρές Fourier

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\left(\frac{2\pi}{T}\right)t}$$

$$a_k = \frac{1}{T} \int_T x(t) e^{-jk\omega_0 t} dt.$$

➤ **Απεριοδικά** σήματα – Μετασχηματισμός Fourier:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$$

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

➤ Μετασχηματισμός Fourier για περιοδικά σήματα:

$$X(j\omega) = 2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\omega - k\omega_0)$$

- Συμβολισμός του Μετασχηματισμού Fourier:

$$X(j\omega) = F\{x(t)\}$$

$$x(t) = F^{-1}\{X(j\omega)\}$$

$$x(t) \leftrightarrow X(j\omega)$$
 π.χ. $1 = F\{\delta(t)\}$ $\delta(t) = F^{-1}\{1\}$ $\delta(t) \leftrightarrow 1$
 $e^{-at} u(t) \leftrightarrow 1/(a+j\omega)$

2.3.3 ΙΔΙΟΤΗΤΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ FOURIER (DTFT) [7]

- **Γραμμικότητα**

Ο Μετασχηματισμός Fourier Διακριτού Χρόνου είναι ένας γραμμικός τελεστής. Αν $X_1(e^{j\omega})$ είναι ο DTFT της $x_1(n)$ και $X_2(e^{j\omega})$ ο DTFT της $x_2(n)$, τότε είναι:

$$ax_1(n) + bx_2(n) \leftrightarrow aX_1(e^{j\omega}) + bX_2(e^{j\omega})$$

- **Μετατόπιση**

Η μετατόπιση μιας ακολουθίας στο χρόνο έχει σαν αποτέλεσμα τον πολλαπλασιασμό του DTFT με ένα μιγαδικό εκθετικό όρο:

$$x(n-n_0) \leftrightarrow e^{-jn_0\omega} X(e^{j\omega})$$

- **Αντιστροφή στο χρόνο**

Η αντιστροφή στο χρόνο μιας ακολουθίας έχει ως αποτέλεσμα την αντιστροφή στη συχνότητα του DTFT

$$x(-n) \leftrightarrow X(e^{-j\omega})$$

- **Διαμόρφωση**

Πολλαπλασιάζοντας μια ακολουθία με ένα μιγαδικό εκθετικό όρο, έχουμε ως αποτέλεσμα μια μετατόπιση στη συχνότητα του DTFT:

$$e^{jn\omega_0} x(n) \leftrightarrow X(e^{j(\omega-\omega_0)})$$

Έτσι, η διαμόρφωση μιας ακολουθίας με ένα σήμα συνημιτόνου στη συχνότητα ω_0 , μετατοπίζει κατά ω_0 το φάσμα στη συχνότητα:

$$x(n) \cos n\omega_0 \leftrightarrow \frac{1}{2} X(e^{j(\omega-\omega_0)}) + \frac{1}{2} X(e^{j(\omega+\omega_0)})$$

- **Το θεώρημα της Συνέλιξης**

Η Συνέλιξη στο πεδίο του χρόνου ισοδυναμεί με την πράξη του πολλαπλασιασμού στο πεδίο της συχνότητας. Ειδικότερα, το θεώρημα αυτό λέει ότι ο DTFT μιας ακολουθίας η οποία

σχηματίζεται από δύο ακολουθίες, $x(n)$ και $h(n)$, είναι το γινόμενο των DTFT των ακολουθιών $x(n)$ και $h(n)$:

$$x(n) * h(n) \leftrightarrow H(e^{j\omega})X(e^{j\omega})$$

- **Το Θεώρημα του Parseval**

Το Θεώρημα του Parseval δηλώνει ότι ο τελεστής DTFT διατηρεί την συνολική ενέργεια κατά την μετάβαση από το πεδίο του χρόνου στο πεδίο της συχνότητας.

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega$$

ΚΕΦΑΛΑΙΟ 3

ARDUINO



3.1 ΟΡΙΣΜΟΣ [10][11]

Το Arduino είναι μία πλακέτα στην οποία βρίσκεται ενσωματωμένος ένας μικροελεγκτής. Είναι ανοιχτού κώδικα με εισόδους-εξόδους όπου μπορούμε να το συνδέσουμε στον Η/Υ και να προγραμματίσουμε από ένα απλό περιβάλλον ανάπτυξης.

Με το Arduino μπορούμε να αναπτύξουμε διαδραστικά αντικείμενα, να δεχτούμε πληθώρα αισθητηρίων οργάνων, διακόπτες αλλά και να ελέγχουμε διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου.

Πολύ σημαντικό είναι ότι να προσθέσουμε 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. Έτσι έχοντας φαντασία, ακόμα και ο πιο αρχάριος χρήστης συνδέοντας τα led, τους αισθητήρες κ.α. μπορεί προγραμματίζοντας να περιηγηθεί στην επιστήμη της φυσικής και να δημιουργήσει κάτι εντελώς δικό του.

3.2 ΙΣΤΟΡΙΚΟ [8]

Το Arduino ξεκίνησε από την πόλη Ιβρέα της Ιταλίας το 2005.

Ο καθηγητής Massimo Banzi έψαχνε έναν ευκολότερο τρόπο για τους φοιτητές του να μπορούν να αναπτύσσουν ενσωματωμένα συστήματα οικονομικά και αποδοτικά αξιοποιώντας τις δυνατότητες και τις ευκαιρίες που μπορεί να προσφέρει το ελεύθερο λογισμικό.

Συζήτησε το πρόβλημα του με τον David Cueartielles, μηχανικός από το πανεπιστήμιο Malmo της Σουηδίας και μαζί αποφάσισαν να δημιουργήσουν ένα μικροελεγκτή.

Το όνομα Arduino δόθηκε από έναν ιστορικό χαρακτήρα, τον Arduin της πόλης Ιβρέα. Το πρώτο Arduino που δημιουργήθηκε ονομάστηκε Serial Arduino.

3.3 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ [12]

Το Arduino βασίζεται στην γλώσσα προγραμματισμού Wiring (πρόκειται για μία γλώσσα παρόμοια με τις γλώσσες C/C++ για μικροελεγκτές AVR όπως ο ATmega και υποστηρίζει όλες τις βασικές δομές της C και C++) καθώς και σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE: Intergrated Development Enviroment) μία εφαρμογή γραμμένη σε Java που τρέχει σε Windows, Linux και MAC OS X (Arduino software).

Το Arduino IDE παρέχει ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων μας που ονομάζονται sketch, αρκετά έτοιμα παραδείγματα, μερικές έτοιμες βιβλιοθήκες για να χειριζόμαστε μέσω του κώδικα πιο εύκολα τα εξαρτήματα που συνδέουμε στο Arduino, τον compiler για την μεταγλώττιση των sketch, ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής της σειριακής θύρας (USB) και τέλος μας επιτρέπει να ανεβάσουμε το μεταγλωττισμένο sketch στο Arduino.

Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Χρησιμοποιούνται ουσιαστικά οι ίδιες εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπους δεδομένων και τους ίδιους τελεστές όπως και στην C. Υπάρχουν όμως και κάποιες άλλες εντολές που βοηθούν στην διαχείριση του hardware του Arduino.

Στον κάτωθι πίνακα παρουσιάζουμε τις πιο σημαντικές:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	(pin, mode)	Καθορίζει αν το συγκεκριμένο

				ψηφιακό <i>pin</i> θα είναι <i>pin</i> εισόδου ή <i>pin</i> εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	-	(<i>pin, pinstatus</i>)	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	-	(<i>type</i>)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς (V_{ref}) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως V_{ref} .
analogWrite	Εντολή	-	(<i>pin, value</i>)	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με <i>value</i> 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την

				στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2^{32} ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
delay	Εντολή	-	(<i>time</i>)	Σταματά προσωρινά την ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως 2^{32}). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	-	(<i>interrupt, function, triggermode</i>)	Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i> , ώστε να ενεργοποιεί την συνάρτηση <i>function</i> , κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i> : <ul style="list-style-type: none"> • LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW) • RISING (όταν από LOW γίνει HIGH) • FALLING (όταν από HIGH γίνει LOW)

				<ul style="list-style-type: none"> • CHANGE (όταν αλλάξει κατάσταση γενικά)
detachInterrupt	Εντολή	-	(<i>interrupt</i>)	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των <i>interrupt</i>
interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των <i>interrupt</i> που διακόπηκε προσωρινά από μια εντολή <i>noInterrupts</i> .
Serial.begin	Μέθοδος κλάσης	-	(<i>datarate</i>)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού <i>interface</i> (σε baud)
Serial.println	Μέθοδος κλάσης	-	(<i>data</i>)	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού <i>interface</i> . Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

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

```

//Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...//
void setup()
{
  //αρχικοποιήσεις//
}
void loop()
{
  //κώδικας//
}

```

Η συνάρτηση `setup()` εκτελείται μια φορά μόνο στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις.

Η συνάρτηση `loop()` περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί.

3.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ARDUINO [11]

- Βασίζεται σε τεχνολογίες ανοιχτού κώδικα, που σημαίνει ότι οποιοσδήποτε μπορεί να το χρησιμοποιήσει αναπτύσσοντας διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Ακόμη οι πιο έμπειροι χρήστες μπορούν να συνεχίσουν με την AVR C που είναι για τον προγραμματισμό των Atmel μικροελεγκτών και η γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino.
- Μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα (Windows, Macintosh, Linux).
- Το περιβάλλον προγραμματισμού του Arduino είναι απλό και ευέλικτο, άρα ενδείκνυται και για αρχάριους χρήστες που ενδιαφέρονται για αλληλεπιδραστικά περιβάλλοντα και συσκευές.
- Χαμηλό κόστος. Οι πλακέτες του Arduino είναι πολύ πιο οικονομικές σε σχέση με άλλες πλακέτες μικροελεγκτών.

3.5 ΕΚΔΟΣΕΙΣ [8]

- Τον Σεπτέμβριο του 2006 ανακοινώθηκε το Arduino Mini.
- Τον Οκτώβρη του 2008 ανακοινώθηκε το Arduino Duemilanove. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328.
- Τον Μάρτιο του 2009 ανακοινώθηκε το Arduino Mega. Είναι βασισμένο στο Atmel ATmega1280.
- Από τον Μάιο του 2011 πάνω από 300,000 Arduino ήταν σε χρήση σε όλο τον κόσμο.
- Τον Ιούλιο του 2012 ανακοινώθηκε το Arduino Leonardo. Είναι βασισμένο στο Atmel ATmega32u4.
- Τον Οκτώβριο του 2012 ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM Cortex-M3.

- Τον Νοέμβριο του 2012 ανακοινώθηκε το Arduino Micro. Είναι βασισμένο στο Atmel ATmega32u4.
- Τον Μάιο του 2013 ανακοινώθηκε το Arduino Robot. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες.
- Τον Μάιο του 2013 ανακοινώθηκε το Arduino Yun. Είναι Βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν wifi που συνδύαζε το Arduino με το Linux.

3.6 ΥΛΙΚΟ [8]

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα *bootloader*, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνετε αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για να μετατρέπει μεταξύ σήματος επιπέδου RS-232 και TTL. Τα τωρινά Arduino προγραμματίζονται μέσω USB, αυτό καθιστάτε δυνατό μέσω της εφαρμογής προσαρμοστικών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείτε με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, πρότυπος προγραμματισμός AVR ISP χρησιμοποιείτε)

Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2.2mm). Διάφορες εφαρμογές ασπίδων plug-in είναι εμπορικώς διαθέσιμα.

Το Arduino nano, και το Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να παρέχει male header pins στο κάτω μέρος του board προκειμένου να συνδέονται σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια

είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί είναι το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Άλλες είναι ηλεκτρικά ισάξια αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι. Κάποιες παραλλαγές είναι τελείως διαφορετικοί επεξεργαστές, με ποικίλα επίπεδα συμβατότητας.

3.7 ΠΛΑΚΕΤΕΣ ARDUINO [8]

Το πρωτότυπο υλισμικό του Arduino κατασκευάζεται από την Ιταλική εταιρία Smart Projects. Κάποιες πλακέτες με την μάρκα του Arduino έχουν σχεδιαστεί από την Αμερικάνικη εταιρία SparkFun Electronics. Δεκαέξι εκδοχές του Arduino Hardware έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα:

1. Το Serial Arduino, προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τεχνολογία ATmega8.
2. Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega8.
3. Το Arduino Mini, μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168.
4. Το Arduino Nano, ένα ακόμα πιο μικρό, USB τροφοδοτούμενη έκδοσή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση).
5. Το LilyPad Arduino, ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted ATmega328.
6. Το Arduino NG, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega8.
7. Το Arduino NG plus, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία atmega168.
8. Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega168.
9. Το Arduino Diecimila, με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε ένα DIP28 πακέτο.
10. Το Arduino Duemilanove (“2009”), χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενης.
11. Το Arduino Mega, χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη.

12. Το Arduino Uno, χρησιμοποιώντας την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilanove, αλλά ενώ το Duemilanove χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας.

13. Το Arduino Mega2560, χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει την νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.

14. Το Arduino Leonardo, με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012.

15. Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση.

16. Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller.

Μερικές από τις αναφερόμενες πλακέτες Arduino παρουσιάζονται και σχηματικά στις κάτωθι εικόνες:



ARDUINO UNO



ARDUINO LEONARDO



ARDUINO MICRO



ARDUINO DUE



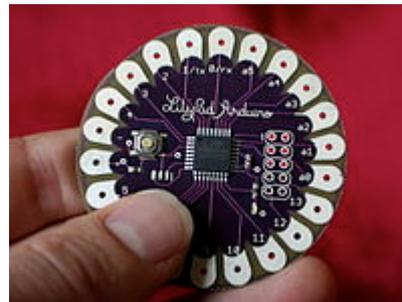
ARDUINO YUN



ARDUINO ESPLORA



ARDUINO ROBOT



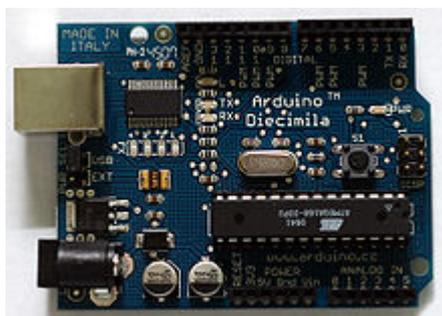
ARDUINO LILYPAD



ARDUINO ETHERNET



ARDUINO TRE



ARDUINO DIECIMILE

3.8 SHIELDS [12]

Τα shields είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες έτσι ώστε να κουμπώνουν πάνω στο Arduino. Μπορούμε να συνδέσουμε πάνω στο Arduino περισσότερα του ενός shield αλλά όχι απεριόριστα αφού το κάθε shield χρησιμοποιεί ορισμένους από τους πόρους συνδεσιμότητας του Arduino. Μερικά shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια Pin του Arduino για επικοινωνία με αυτό. Επίσης, επειδή κάποια shield δεν προωθούν τις συνδέσεις του Arduino (όπως π.χ. οι οθόνες οι οποίες δεν έχουν νόημα αν τις καλύψουμε από πάνω με ένα επόμενο shield), υπάρχουν ειδικά extender shield που κουμπώνουν στο Arduino και δίνουν την δυνατότητα σε δύο άλλα shield να κουμπώσουν πάνω τους, λειτουργώντας σαν πολύπριζα.

Όπως και για το ίδιο το Arduino, το βασικό πλεονέκτημα των shield δεν είναι τόσο το προφανές πλεονέκτημα του έτοιμου hardware όσο ότι συνοδεύονται συνήθως από έτοιμες βιβλιοθήκες που μας επιτρέπουν να προγραμματίζουμε τα sketch σας σε high level. Έτσι, λόγω χάρη, δεν χρειάζεται να διαβάζουμε datasheet ή να γίνουμε ηλεκτρονικοί για να συνδέσουμε και να λειτουργήσουμε ένα GPS module πάνω στο Arduino. Απλά συνδέουμε το shield, εγκαθιστούμε τη βιβλιοθήκη που το συνοδεύει και χρησιμοποιούμε μια έτοιμη συνάρτηση -του στυλ getLocation- για να πάρουμε το γεωγραφικό στίγμα και να το επεξεργαστούμε περαιτέρω στο sketch μας.

Τα shield μας λύνουν τα χέρια όταν θέλουμε να δημιουργήσουμε εύκολα ένα πραγματικά πρακτικό project.



ARDUINO GSM SHIELD



ARDUINO ETHERNET SHIELD



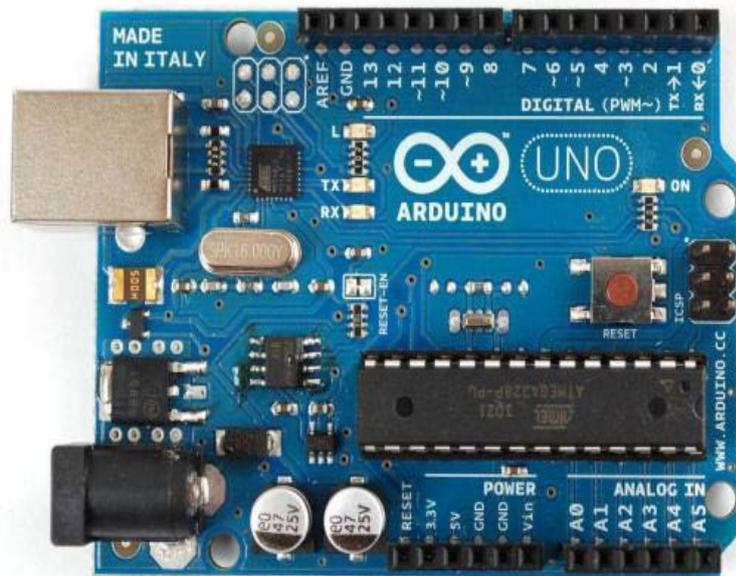
ARDUINO WIFI SHIELD



ARDUINO MOTOR SHIELD

3.9 ARDUINO UNO

Για την παρούσα πτυχιακή εργασία χρησιμοποιήσαμε την κάτωθι πλατφόρμα Arduino Uno η οποία περιέχει όλα όσα χρειάζονται για να υποστηρίξουν τον μικροελεγκτή, καθώς και ένα καλώδιο USB για να συνδέσουμε την πλακέτα με τον υπολογιστή.



Πλακέτα ARDUINO UNO



USB καλώδιο

3.9.1 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO UNO [9]

Μικροελεγκτής	ATmega328
Τάση λειτουργίας	5V
Τάση εισόδου (προτεινόμενη)	7-12V
Τάση εισόδου (όρια)	6-20V
Ψηφιακοί ακροδέκτες I/O	14 (of which 6 provide PWM output)
Ψηφιακοί ακροδέκτες εισόδου	6
DC ρεύμα ανά I/O ακροδέκτη	40 mA
DC ρεύμα για 3.3V ακροδέκτη	50 mA
Μνήμη Flash	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Ταχύτητα ρολογιού	16 MHz

3.9.2 ΜΝΗΜΗ [9]

Το ολοκληρωμένο ATmega328 έχει 32KB μνήμη flash εκ των οποίων τα 0,5 KB χρησιμοποιούνται από τον bootloader. Έχει επίσης 2KB SPAM και 1KB EEPROM (τα οποία μπορούν να διαβαστούν και να γραφούν με την βιβλιοθήκη EEPROM).

3.9.3 ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ [9]

Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες του Arduino, μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος με την χρήση των συναρτήσεων `pinMode()`, `digitalWrite()` και `digitalRead()`. Λειτουργούν στα 5 Volts και ο κάθε ακροδέκτης μπορεί να παρέχει ή να λαμβάνει το μέγιστο 40 mA και έχει ένα εσωτερικό pull up αντιστάτη των 20-50 kohms.

Επιπροσθέτως κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες:

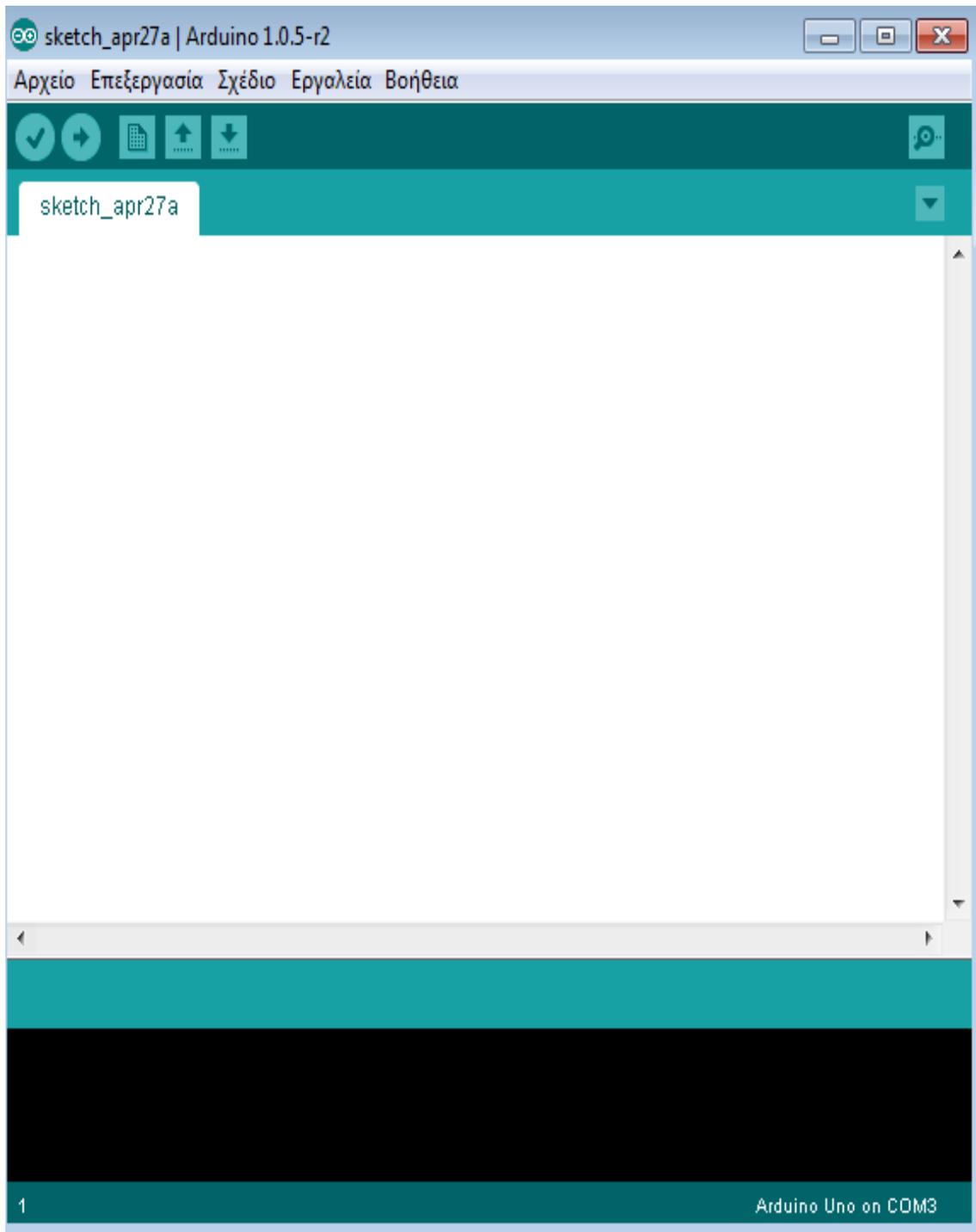
- Σειριακή Λειτουργία: 0 (RX) and 1 (TX) : Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τους αντίστοιχους του ολοκληρωμένου ATmega8U2 USB-to-TTL Serial chip.
- Εξωτερικές Διακοπές 2 και 3 : Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης χρησιμοποιώντας την συνάρτηση attachInterrupt().
- PWM 3, 5, 6, 9, 10 και 11 : Παρέχουν έξοδο 8-bit PWM με την συνάρτηση analogWrite().
- SPI 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) : Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI χρησιμοποιώντας την βιβλιοθήκη SPI.
- LED 13 : Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED, όπου όταν ο ακροδέκτης έχει τιμή HIGH, το LED είναι αναμμένο και όταν έχει την τιμή LOW είναι αντίστοιχα σβηστό.

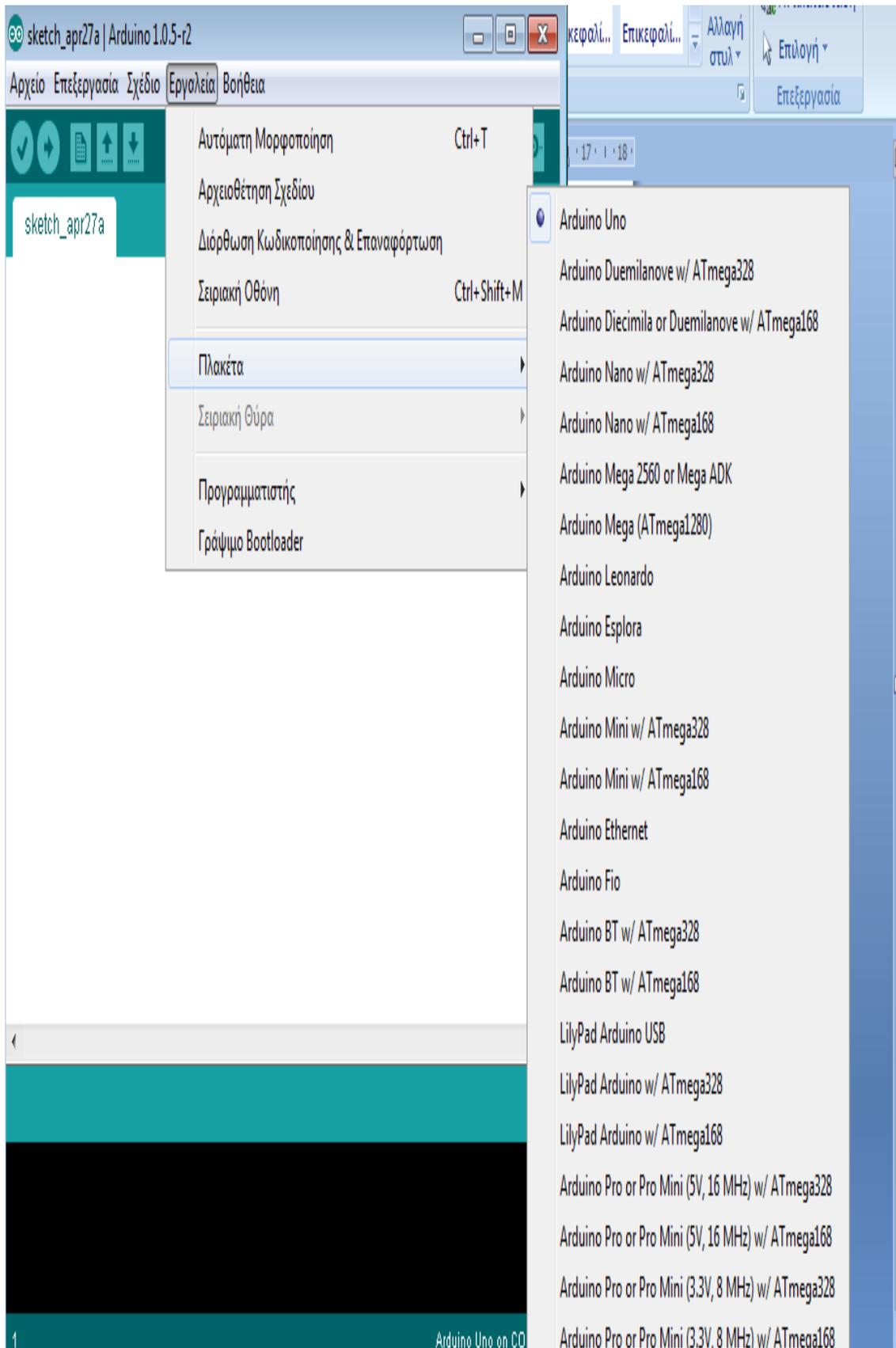
Το Arduino UNO έχει 6 αναλογικές εισόδους, ονομαζόμενες από A0 έως A5 οι οποίες έχουν κάποιες άλλες λειτουργίες:

- TWI A4 ή ακροδέκτης SDA και A5 ή ακροδέκτης SCL : Ο ακροδέκτης αυτός υποστηρίζει την TWI επικοινωνία χρησιμοποιώντας την βιβλιοθήκη Wire.
- AREF : Αναφερόμενη τάση για αναλογικές εισόδους Χρησιμοποιείται με την συνάρτηση analogReference().
- Reset. Αν τεθεί σε κατάσταση LOW τότε επαννεκινεί τον Μικροελεγκτή.

3.9.4 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ARDUINO UNO

Αφού εγκαταστήσαμε με επιτυχία την δωρεάν έκδοση του Arduino UNO 1.0.5-r2 από την ιστοσελίδα <http://arduino.cc/en/Main/Software#toc2> παρουσιάζουμε τις παρακάτω εικόνες όπου παραθέτουν το περιβάλλον ανάπτυξης του Arduino που εργαστήκαμε :

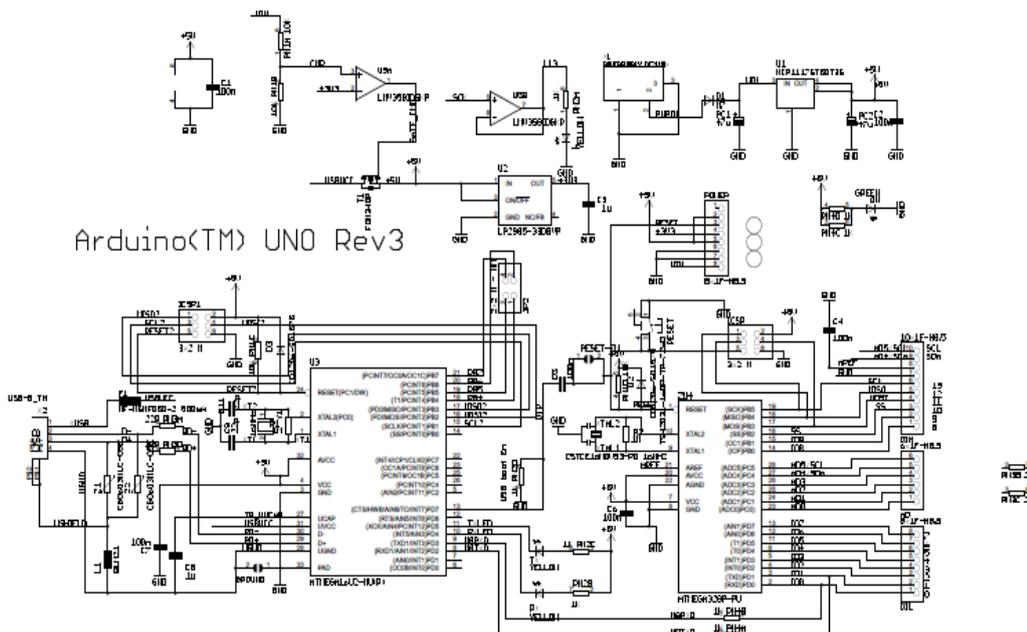




3.9.5 Βασικές λειτουργίες του IDE : [10]

	Έλεγχος του κώδικα για λάθη.
	Τερματισμός της σειριακής κονσόλας.
	Δημιουργία νέου έργου (sketch)
	Παρουσίαση μενού με όλα τα αποθηκευμένα έργα. Πατώντας σε ένα απο αυτά ανοίγει για επεξεργασία.
	Αποθήκευση του έργου.
	Μεταγλώττιση του κώδικα και ανέβασμά του στο Arduino.
	Εμφάνιση της σειριακής κονσόλας. Αποστολή και λήψη δεδομένων που στάλθηκαν μέσω της σειριακής θύρας.

3.9.6 ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ARDUINO UNO Rev3 [13]



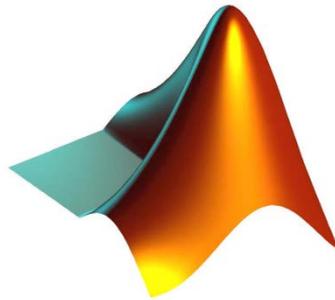
Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

ΚΕΦΑΛΑΙΟ 4

MATLAB



4.1 ΟΡΙΣΜΟΣ MATLAB [14][20]

Το MATLAB είναι ένα διαδραστικό πρόγραμμα που περιέχει αριθμητικούς υπολογισμούς με δυνατότητες προγραμματισμού που το καθιστούν ως ένα ισχυρότατο εργαλείο στην υλοποίηση προσομοιώσεων και μοντελοποιήσεων συστημάτων, στην έρευνα, στην επιστήμη των μηχανικών και στις επικοινωνίες. Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων MATrix LABoratory (Εργαστήριο πινάκων).

Σε αντίθεση με τα λογισμικά Maple και Mathematica, το MATLAB στις αρχικές του εκδοχές δεν έκανε συμβολικούς υπολογισμούς. Στις νεότερες εκδοχές του, το πακέτο περιλαμβάνει εργαλείοθήκες που επιτρέπουν συμβολικούς υπολογισμούς. Η βασική του διαφορά με τα ανωτέρω λογισμικά (Maple – Mathematica) είναι ότι το MATLAB επιλύει προβλήματα σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic), δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος.

Όπως υποδηλώνει το όνομά του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικών πινάκων κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων).

4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ MATLAB [15][17]

- Το περιβάλλον του είναι φιλικό προς τον χρήστη.
- Παρέχει άμεσες δυνατότητες γραφικής απεικόνισης.
- Έχει πληθώρα ενσωματωμένων συναρτήσεων.
- Παρέχει τη δυνατότητα προσθήκης συναρτήσεων γραμμένων από τον χρήστη.
- Ο προγραμματισμός στο MATLAB είναι απλός.
- Περιλαμβάνει πληθώρα εργαλείων από διάφορες επιστημονικές περιοχές.
- Υψηλή απόδοση και ταχύτητα υπολογιστικών αναλύσεων.
- Παρέχει την δυνατότητα προσομοίωσης φυσικών συστημάτων.
- Παρέχει την δυνατότητα αμφίδρομης επικοινωνίας με πληθώρα άλλων προγραμμάτων και εφαρμογών.
- Παρέχει την δυνατότητα υλοποίησης αλγορίθμων.

4.3 ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ [18]

Στο MATLAB θα συναντήσουμε τα παράθυρα :

Command Window : Περιέχει την γραμμή εντολών, όπου μετά την προτροπή >>, πληκτρολογούμε τις εντολές που θέλουμε να εκτελέσουμε.

Σημαντικές παρατηρήσεις:

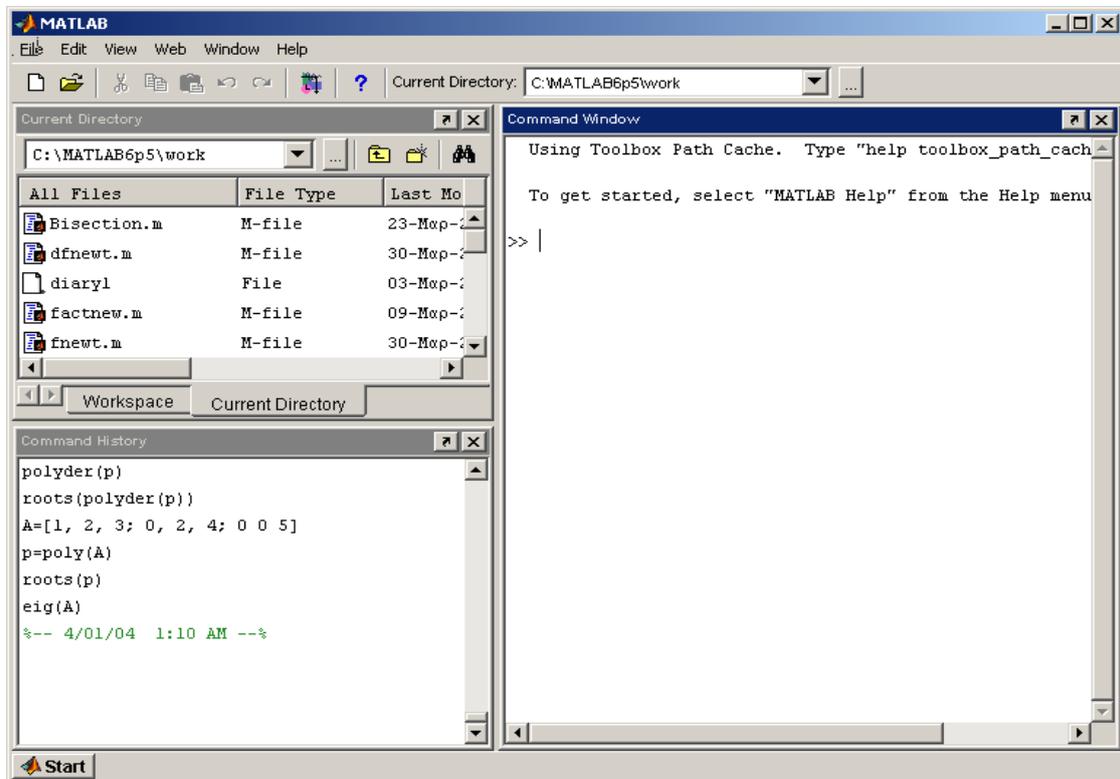
- Μπορούμε να γράψουμε περισσότερες από μια εντολές σε μια γραμμή τις οποίες χωρίζουμε είτε με κόμμα, είτε με ερωτηματικό (αν δεν θέλουμε να τυπωθεί το αποτέλεσμα στο παράθυρο εντολών).
- Στο MATLAB υπάρχει διάκριση μεταξύ μικρών και κεφαλαίων γραμμάτων.
- Όταν γράψουμε το όνομα μιας μεταβλητής, το MATLAB τυπώνει στην οθόνη την τιμή της.
- Πατώντας τα πλήκτρα με τα πάνω και κάτω βέλη ([↑] και [↓]) μπορούμε να διατρέξουμε όλες τις προηγούμενες εντολές. Επίσης μια προηγούμενη εντολή μπορεί να επαναληφθεί αν γράψουμε τα πρώτα γράμματα και μετά πατήσουμε το πλήκτρο με το πάνω βέλος [↑].

Current directory : Δείχνει τον τρέχοντα φάκελο και τα αρχεία που εμφανίζονται σε αυτόν.

Command History : εμφανίζονται οι εντολές που εκτελέστηκαν όταν έγινε η εκκίνηση του προγράμματος, αλλά και οι εντολές που πληκτρολογήθηκαν τις προηγούμενες φορές.

Workspace : Αποθηκεύονται όλες οι μεταβλητές και οι πίνακες που δημιουργούνται μέσω των εντολών που δίνουμε στο command window.

Κατά τη διάρκεια μιας εργασίας στο MATLAB μπορεί να εμφανιστούν αυτόματα και άλλα παράθυρα όταν αυτό απαιτείται όπως **παράθυρα κειμένου** (document windows), **παράθυρα γραφικών** (graphics windows) και **παράθυρα σύνταξης αρχείων** (editing windows).



Υπάρχουν δύο εναλλακτικοί τρόποι που μπορούμε να εργαστούμε στο MATLAB:

- 1) Μέσω του command window όπου μπορούμε να δώσουμε μεμονωμένα κάποιες εντολές και να πάρουμε άμεσα τα αποτελέσματα ή
- 2) Μέσω scripts (προγραμμάτων), όπου τα γράφουμε στο περιβάλλον του editor και εκτελούνται είτε απευθείας από τον editor δίνοντας την εντολή Run, είτε από το command window γράφοντας το όνομα του script. Τα scripts έχουν προέκταση .m και συνήθως αποθηκεύονται στο αρχείο work των directories του MATLAB.

4.4 ΧΡΗΣΙΜΕΣ ΕΝΤΟΛΕΣ ΣΤΟ MATLAB [14]

ΕΝΤΟΛΗ	ΕΡΜΗΝΕΙΑ
help	βοήθεια
what	κατάλογος των αρχείων MATLAB στον τρέχοντα φάκελο ομαδοποιημένων σύμφωνα με τον τύπο τους
who – whos	κατάλογος ενεργών μεταβλητών εργασίας
load	φόρτωση από αρχείο των μεταβλητών εργασίας
save	αποθήκευση σε αρχείο των μεταβλητών εργασίας
exit ή quit	έξοδος από το πρόγραμμα
clear	διαγραφή ενεργών μεταβλητών
diary	αποθήκευση εργασίας σε αρχείο
clc	καθαρισμός παραθύρου εργασίας
disp	απεικόνιση μεταβλητών στην οθόνη
format	μορφή εκτύπωσης μεταβλητών στην οθόνη
input	υποβολέας για είσοδο δεδομένων

4.5 ΑΠΛΕΣ ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ [18]

Στον κάτωθι πίνακα παραθέτουμε τις πράξεις που μπορεί να εκτελέσει το MATLAB, καθώς και παραδείγματα εντολών με το αποτέλεσμά τους : [20]

ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ	ΕΝΤΟΛΗ	ΑΠΟΤΕΛΕΣΜΑ
Πρόσθεση (+)	>> 6+2	ans = 8
Αφαίρεση (-)	>> 5-3	ans = 2

Πολλαπλασιασμός (*)	>> 4*5	ans = 20
Δεξιά διαίρεση (/)	>> 2/4	ans = 0.5
Αριστερή διαίρεση (\)	>> 2\4	ans = 2
Ύψωση σε δύναμη (^)	>> 2^3	ans = 8

Το διπλό σύμβολο >> , δηλώνει ότι το πρόγραμμα είναι έτοιμο να δεχθεί για εκτέλεση την εντολή.

Αφού γράψουμε την εντολή, χρησιμοποιώντας πάντοτε αριθμητικούς τελεστές που δηλώνουν τις πράξεις μεταξύ των αριθμών, εκτελούμε την πράξη με το πάτημα του πλήκτρου Enter ,

π.χ. >> 6+2 Enter

Μετά την εκτέλεση της εντολής, εμφανίζεται στην οθόνη η μεταβλητή ans στην οποία αποθηκεύεται αυτόματα η τελευταία απάντηση που δόθηκε.

ans =
8 ,

Τονίζουμε, επίσης, ότι το matlab, εκτός από τον τελεστή διαίρεσης από τα αριστερά, διαθέτει και το τελεστή διαίρεσης από τα δεξιά. Σε κάθε περίπτωση το αποτέλεσμα διαφέρει.

4.6 ΣΥΝΑΡΤΗΣΕΙΣ ΣΤΟ MATLAB ΚΑΙ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥΣ [18]

ΣΥΝΑΡΤΗΣΗ	ΛΕΙΤΟΥΡΓΙΑ
sin	ημίτονο
sinh	υπερβολικό ημίτονο
asin	τόξο ημιτόνου

asinh	τόξο υπερβολικού συνημίτονου
cos	συνημίτονο
cosh	υπερβολικό συνημίτονο
acos	τόξο συνημίτονου
acosh	τόξο υπερβολικού συνημίτονου
tan	εφαπτομένη
tanh	τόξο υπερβολικής εφαπτομένης
atan	τόξο εφαπτομένης
atanh	αντίστροφη υπερβολική εφαπτομένη
sec	τέμνουσα
sech	υπερβολική τέμνουσα
asec	αντίστροφη τέμνουσα
asech	αντίστροφη υπερβολική τέμνουσα
csc	συντέμνουσα
csch	υπερβολική συντέμνουσα
acsc	αντίστροφη συντέμνουσα
acsch	αντίστροφη υπερβολική συντέμνουσα
cot	συνεφαπτομένη
coth	υπερβολική συνεφαπτομένη
acot	αντίστροφη συνεφαπτομένη
acoth	αντίστροφη υπερβολική συνεφαπτομένη
exp	εκθετική συνάρτηση e^x
log	λογάριθμος
log10	δεκαδικός λογάριθμος
sqrt	τετραγωνική ρίζα
abs	απόλυτη τιμή
angle	γωνίες φάσης στοιχείων μιγαδικού

conj	συζυγής μιγαδικού
imag	φανταστικό μέρος μιγαδικού
real	πραγματικό μέρος μιγαδικού
fix	ακέραιο μέρος
floor	κάτω ακέραιο μέρος
ceil	πάνω ακέραιο μέρος
round	στρογγυλοποίηση
rem	υπόλοιπο διαίρεσης
mod	προσημασμένο υπόλοιπο διαίρεσης
sign	πρόσημο

4.7 ΕΙΔΙΚΕΣ ΣΤΑΘΕΡΕΣ ΚΑΙ ΜΕΤΑΒΛΗΤΕΣ [18]

ΣΤΑΘΕΡΑ ή ΜΕΤΑΒΛΗΤΗ	ΛΕΙΤΟΥΡΓΙΑ
pi	η σταθερά π
i, j	φανταστική μονάδα ($i = \sqrt{-1}$)
inf	άπειρο ∞
-inf	μείον άπειρο $-\infty$
eps	η μεγαλύτερη ακρίβεια πράξης που μπορούμε να έχουμε
nan	μη αριθμός
realmax	ο μεγαλύτερος δεκαδικός αριθμός που ορίζεται
realmin	ο μικρότερος δεκαδικός αριθμός που ορίζεται

4.8 ΔΙΑΝΥΣΜΑΤΑ

4.8.1 ΔΙΑΝΥΣΜΑΤΑ ΓΡΑΜΜΗΣ [18]

Τα στοιχεία που αποτελούν ένα διάνυσμα γραμμής περικλείονται από αγκύλες []. Τα στοιχεία αυτά ξεχωρίζουν μεταξύ τους είτε με κενά, είτε με κόμματα. Τα στοιχεία του διανύσματος προσδιορίζονται με δείκτες, όπου ο δείκτης που δείχνει το πρώτο στοιχείο του διανύσματος έχει την τιμή 1. Αναφερόμαστε σε κάποια θέση του διανύσματος χρησιμοποιώντας παρενθέσεις (). Μπορούμε να τροποποιήσουμε ένα μόνο στοιχείο του διανύσματος, δίνοντας τιμή μόνο στο συγκεκριμένο στοιχείο.

Παραδείγματα :

	ΕΝΤΟΛΗ	ΑΠΟΤΕΛΕΣΜΑ
Διάνυσμα γραμμής 6 στοιχείων με όνομα a	a=[5 9 6 7 2 1] ή a=[5,9,6,7,2,1]	ans= 5 9 6 7 2 1
Το τρίτο στοιχείο του διανύσματος a	a(3)	ans= 6
Καταχώρηση της τιμής 36 στην 3 θέση του διανύσματος	a(3)=36	ans= 5 9 36 7 2 1
Καταχώρηση της τιμής 59 στην 9 θέση του διανύσματος. Βάζουμε 0 στις θέσεις που δεν έχουν οριστεί.	a(9)=59	ans= 5 9 36 7 2 1 0 0 59

Εάν όμως έχουμε διανύσματα μεγάλου μεγέθους τα παραδείγματα του ανωτέρω πίνακα δεν μας εξυπηρετούν. Έτσι αν τα στοιχεία του διανύσματος βρίσκονται σε ίσες αποστάσεις μεταξύ τους μπορούμε να δηλώσουμε το διάνυσμα αυτό ως εξής :

	ΕΝΤΟΛΗ	ΑΠΟΤΕΛΕΣΜΑ
Διάνυσμα a που αποτελείται από στοιχεία που αρχίζουν στο 1 και τελειώνουν στο 9 με βήμα 2	a=1:2:9	ans= 1 3 5 7 9

4.8.2 ΔΙΑΝΥΣΜΑΤΑ ΣΤΗΛΗΣ [18]

Και σε αυτήν την περίπτωση τα στοιχεία που αποτελούν το διάνυσμα περικλείονται από αγκύλες [] με την μόνη διαφορά ότι τα στοιχεία μεταξύ τους διαχωρίζονται με ερωτηματικά.

Παραδείγματα :

	ΕΝΤΟΛΗ	ΑΠΟΤΕΛΕΣΜΑ
Διάνυσμα στήλης 4 στοιχείων με όνομα a	a=[5; 9; 6; 7] ή a=[5 9 6 7]	ans= 5 9 6 7

4.8.3 ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ ΔΙΑΝΥΣΜΑΤΩΝ

Με τα διανύσματα μπορούμε να εκτελέσουμε τις πράξεις :

- Πρόσθεση – αφαίρεση (Σημαντική προϋπόθεση : Τα διανύσματα να έχουν το ίδιο μέγεθος).
- Πολλαπλασιασμός – διαίρεση (Γίνεται η πράξη στοιχείο προς στοιχείο χρησιμοποιώντας τον τελεστή ./ π.χ.a./b).
- Ύψωση σε δύναμη.

4.8.4 ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΝΥΣΜΑΤΩΝ

- **length** : Επιστρέφει το μήκος του διανύσματος.
- **max** : Επιστρέφει Την τιμή του μέγιστου στοιχείου του διανύσματος.
- **min** : Επιστρέφει την τιμή του ελάχιστου στοιχείου του διανύσματος.

- **sum** : Επιστρέφει το άθροισμα των στοιχείων του διανύσματος.
- **prod** : Επιστρέφει το γινόμενο στοιχείων του διανύσματος.
- **norm** : Επιστρέφει την τετραγωνική ρίζα του αθροίσματος των στοιχείων του διανύσματος υψωμένα στο τετράγωνο.
- **sort** : Επιστρέφει τα στοιχεία του διανύσματος ταξινομημένα.
- **find** : Επιστρέφει τις θέσεις των στοιχείων του διανύσματος τα οποία ικανοποιούν μια συνθήκη.
- **mean** : Επιστρέφει τον μέσο όρο των στοιχείων του διανύσματος.
- **median** : Επιστρέφει την διάμεσο τιμή του διανύσματος.
- **diff** : Επιστρέφει την αφαίρεση ενός στοιχείου από το αμέσως επόμενο του.

4.9 ΓΡΑΦΗΜΑΤΑ ΣΤΟ MATLAB

Το MATLAB έχει εξαιρετικές δυνατότητες για γραφικά και είναι εφοδιασμένη με αρκετές συναρτήσεις για εύκολο και ευέλικτο σχεδιασμό επίπεδων καμπυλών, τρισδιάστατων επιφανειών, ισούψων, παραμετρικών δισδιάστατων αλλά και τρισδιάστατων καμπυλών κα.

4.9.1 ΣΥΝΑΡΤΗΣΕΙΣ ΓΡΑΦΗΜΑΤΩΝ [16]

Οι κυριότερες συναρτήσεις για το γραφικό περιβάλλον είναι:

ΚΩΔΙΚΟΣ	ΛΕΙΤΟΥΡΓΙΑ	ΠΑΡΑΔΕΙΓΜΑ
plot	Δημιουργία γραφήματος	plot(x,y)
subplot	Διαιρεί την γραφική παράσταση σε υπογραφικές παραστάσεις	subplot(mnr)
title	Προσθήκη τίτλου	title('Titlos')
xlabel	Προσθήκη ετικέτας στον οριζόντιο άξονα	xlabel('Xronos, t')
ylabel	Προσθήκη τίτλου στον κατακόρυφο άξονα	ylabel('Taxuthta, cm/s')
legend	Προσθήκη λεζάντας	legend('First', 'Second')
text	Προσθήκη κειμένου	Text(x,y 'string')
grid	Δημιουργία πλέγματος	grid grid on grid off
figure	Άνοιγμα άλλου παραθύρου γραφικών	figure(2)

hold	Πάγωμα του τρέχοντος παραθύρου γραφικών για τον σχεδιασμό και άλλων γραφικών	hold on/hold off
axis	Κλείδωμα/ξεκλείδωμα αξόνων Ίσες μονάδες αξόνων Διαγραφή αξόνων Όρια αξόνων.	axis axis equal axis off axis([xmin, xmax,ymin,ymax])

Υπάρχει η δυνατότητα να ορίσουμε εμείς τα χρώματα, τα σύμβολα για τα σημεία και τον τύπο της γραμμής που τα ενώνει. Αυτό επιτυγχάνεται με την εισαγωγή ακόμη ενός ορίσματος στην συνάρτηση plot, μετά από το κάθε ζεύγος διανυσμάτων που έχει σχεδιαστεί. Το νέο όρισμα είναι μία σειρά από ειδικούς χαρακτήρες μέσα σε αποστρόφους. Οι παρακάτω πίνακες εξηγούν όσα προαναφέρθηκαν : [16][19][20]

[COLOR]	COLOR	ΧΡΩΜΑ
b	blue	μπλε
g	green	πράσινο
r	red	κόκκινο
c	cyan	κυανό
m	magenta	μοβ
y	yellow	κίτρινο
k	black	μαύρο
w	white	άσπρο

Για το τύπο του συμβόλου έχουμε τις εξής επιλογές : [16][19]

[STYPE]	SYMBOL	ΣΥΜΒΟΛΟ
.	point	τελεία
o	circle	κύκλος

x	x-mark	χ
+	plus	συν
*	star	αστερίσκος
s	square	τετράγωνο
d	diamond	ρόμβος
v	triangle (down)	κάτω τρίγωνο
^	triangle (up)	άνω τρίγωνο
<	triangle (left)	αριστερό τρίγωνο
>	triangle (right)	δεξιό τρίγωνο
p	pentagram	πεντάλφα
h	hexagram	εξάλφα

Για τον τύπο [ltype] της γραμμής έχουμε τις εξής επιλογές : [16][19]

Ltype	Line type	Τύπος γραμμής
-	solid	συνεχής
:	dotted	λεπτή διακεκομμένη
--	dashed	αδρή διακεκομμένη
-.	dashdot	διακεκομμένη-τελεία

Σημειώνεται ότι δεν είναι απαραίτητο να προσδιορίσουμε υποχρεωτικά το χρώμα και τον τύπο γραμμής. Το matlab χρησιμοποιεί τις προεπιλογές :

- Για το χρώμα [color] το b (μπλε).
- Για τον τύπο [type] το - (συνεχής γραμμή).

Παραδείγματα :

>> plot (x, y, 'g--') : δίνει πράσινη αδρή διακεκομμένη γραμμή.

>> plot (x, y, 'm:') : δίνει μοβ λεπτή διακεκομμένη γραμμή.

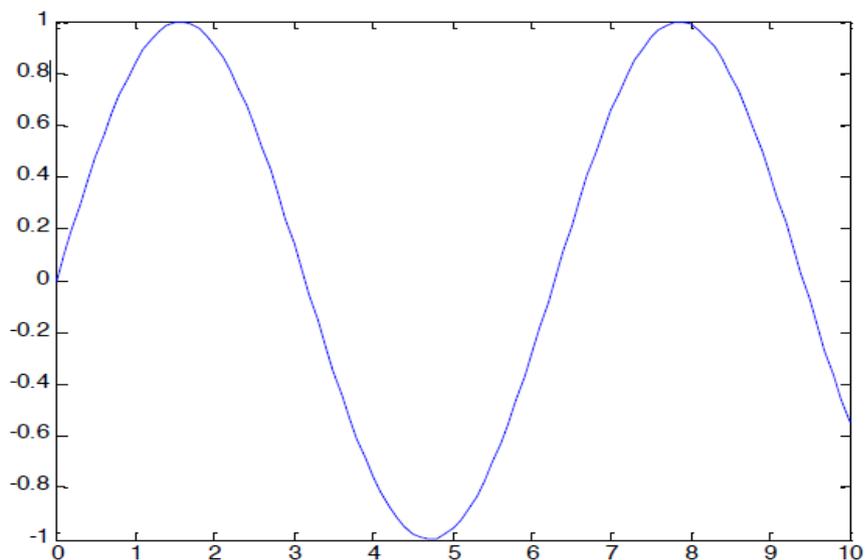
>> plot (x, y, 'c+') : δίνει μια κυανή λεπτή διακεκομμένη γραμμή και το σύμβολο + σε κάθε σημείο.

>> plot (x, y, 'g') : δίνει πράσινη συνεχή γραμμή.

>> plot (x, y, 'o') : δίνει μπλε κύκλους σε κάθε σημείο.

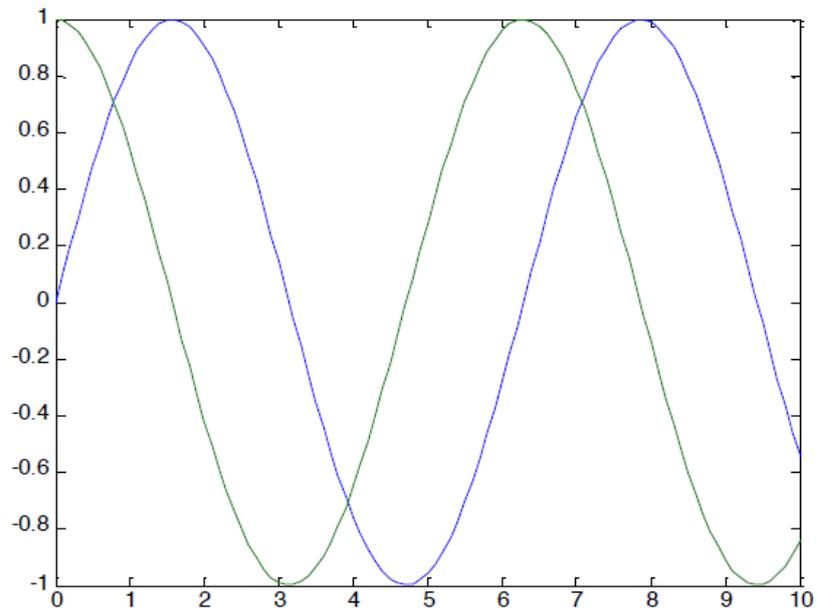
Ας δούμε κάποιες γραφικές παραστάσεις χρησιμοποιώντας όσα προαναφερθήκαν : [21]

Παράδειγμα 1 : >>a=0:0.1:10;
>>b=sin(a);
>>plot(a,b)



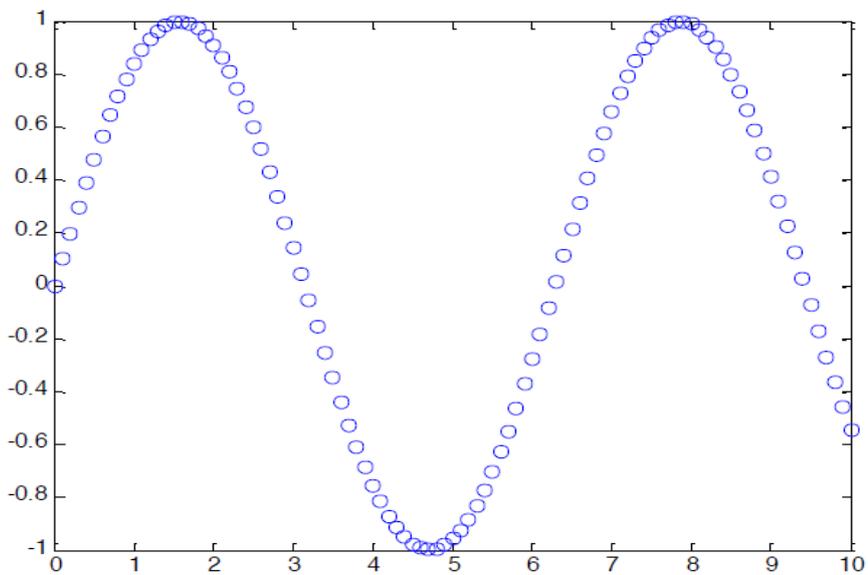
Γράφημα 1 : Δισδιάστατη γραφική όπου τα στοιχεία του διανύσματος b τυπώνονται συναρτήσει των στοιχείων του διανύσματος a.

Παράδειγμα 2 : `>>c=cos(a);`
 `>>plot(a,b,a,c)`



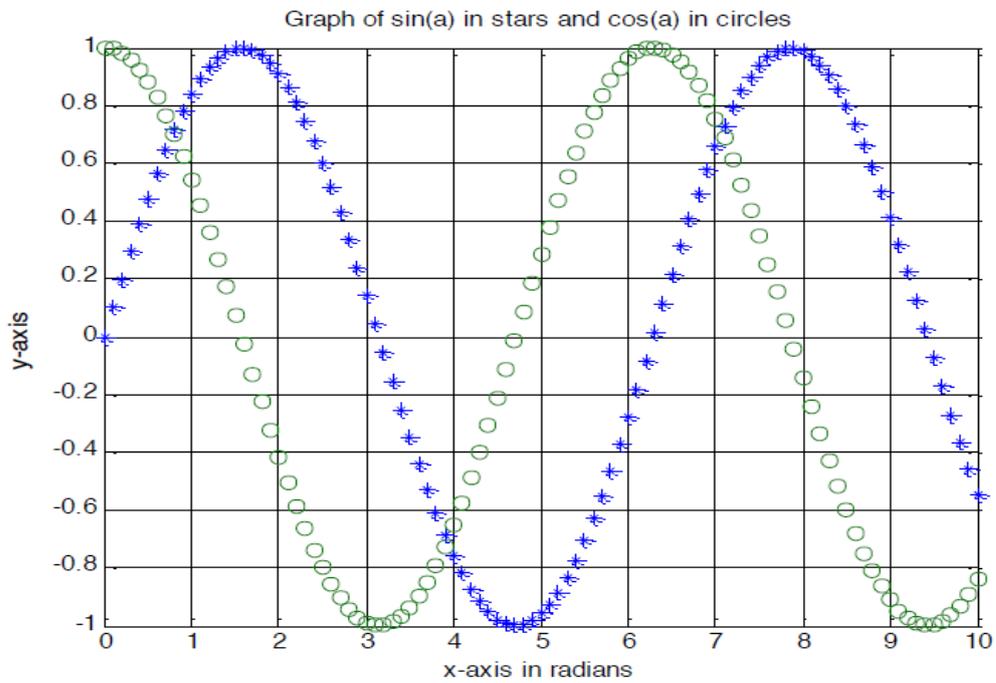
Γράφημα 2 : Στο γράφημα 1 προστέθηκε άλλο ένα διάνυσμα

Παράδειγμα 3 : `>>plot(a,b,'o')`



Γράφημα 3 : Ο τύπος του συμβόλου είναι κύκλος

Παράδειγμα 4 : `>>plot(a, b,'*', a, c, 'o');`
`>>xlabel('x-axis in radians');`
`>>ylabel("y-axis);`
`>>title('Graph of sin(a) in stars and cos(a) in circles');`
`>>grid;`



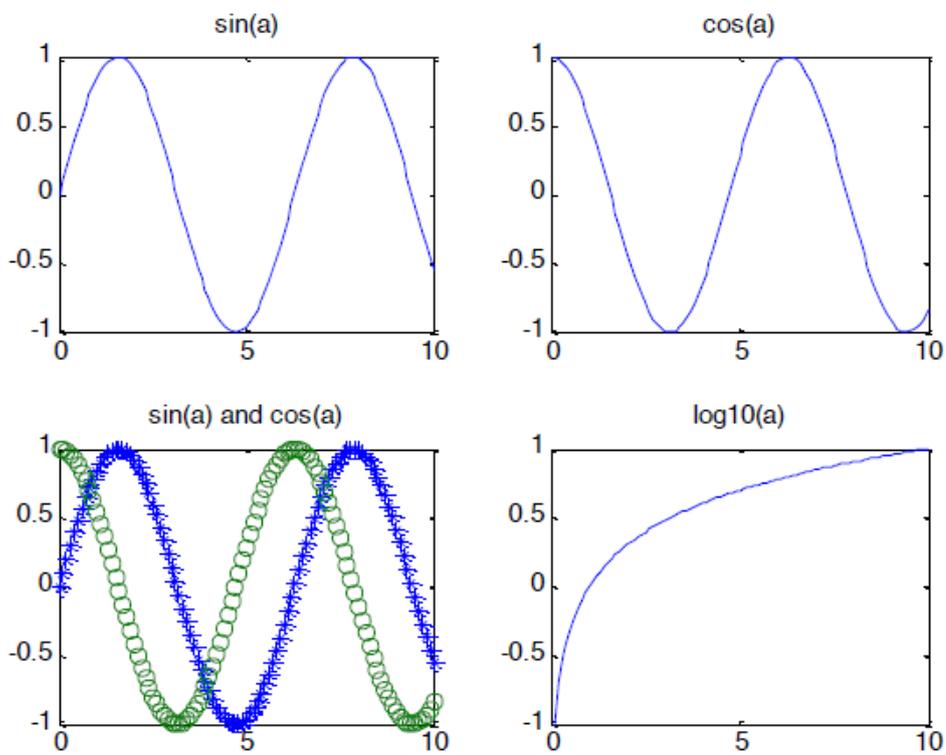
Γράφημα 4

Παράδειγμα 5 :

```

>>subplot(221), plot(a, b)
>>title('sin(a)')
>> subplot(222), plot(a, c)
>>title('cos(a)')
>> subplot(223), plot(a, b, '*', a, c, 'o')
>>title('sin(a) and cos(a)')
>> subplot(224), plot(a, log10(a))
>>title('log10(a)')

```



Γράφημα 5

Κ Ε Φ Α Λ Α Ι Ο 5

ΚΩΔΙΚΑΣ ARDUINO

5.1 ΕΙΣΑΓΩΓΗ

Ο κώδικας που ακολουθεί είναι γραμμένος στο περιβάλλον ανάπτυξης IDE του Arduino σε γλώσσα C όπου αναλύουμε το αναλογικό μας σήμα σε πραγματικό χρόνο εφαρμόζοντας το μετασχηματισμό Fourier διακριτών σημάτων (DFT) με τη μέθοδο FFT (Fast Fourier Transform). Το φάσμα στέλνεται σε ηλεκτρονικό υπολογιστή μέσω σειριακής θύρας.

5.2 ΚΩΔΙΚΑΣ

```
#include <stdint.h>
#include <fft.h>
int16_t input[FFT_N]; //τύπος ακέραιας μεταβλητής με μήκος 16bit. Ο πίνακας με
                        //όνομα input έχει το αρχικό σήμα στο χρόνο. Σε αυτόν τον
                        //πίνακα θα αποθηκεύσουμε τα δείγματα.

complex_t fft_buff[FFT_N]; // είναι μιγαδικός και χρειάζεται γιατί ο μετασχηματισμός
                            //Fourier παράγει μιγαδικούς. Ο πίνακας με όνομα fft_buff
                            //είναι ο ενδιάμεσος πίνακας και έχει τους μιγαδικούς του
                            //μετασχηματισμού.

uint16_t output[FFT_N/2]; // τύπος ακέραιας μεταβλητής με μήκος 16bit, δεν έχει
                           //πρόσημο –είναι θετικός διότι έχει το πλάτος του τελικού
                           //φάσματος. Ο πίνακας με όνομα output είναι ο τελικός
                           //πίνακας με το φάσμα πλάτους.

void setup() // Τρέχει μια φορά στην αρχή και εκεί απλά αρχικοποιεί τη
             //σειριακή θύρα με τιμή ταχύτητας (baudrate) 9600.

{
    Serial.begin(9600); // Τιμή ταχύτητας (baudrate) 9600 σειριακής θύρας.
```

```

pinMode(5, OUTPUT);           // Ορίζουμε το pin 5 ως έξοδο.

analogWrite(5, 127);         // Παράγουμε στο pin 5 τετραγωνική κυματομορφή 1KHz.
}

void loop()                  // Τρέχει συνεχώς σαν βρόχος while() οπότε το παράδειγμα
                               παίρνει μια αναλογική τιμή, τη γράφει στη σειριακή
                               θύρα περιμένει 1 δευτερόλεπτο (delay(1000)) και κάνει το
                               ίδιο ξανά και ξανά

{
int i;
for(i=0; i<=FFT_N; i++)      // Το i θα πάρει τις τιμές από 0 έως 63. Η μεταβλητή FFT_N
                               έχει δηλωθεί στο αρχείο fft.h να έχει την τιμή 64
{
input[i]=analogRead(A0);     // διαβάζει την τιμή από την αναλογική είσοδο A0 και την
                               αποθηκεύει σε έναν πίνακα με όνομα i
}

fft_input(input, fft_buff);   // Συνάρτηση μετασχηματισμού FFT που μας δίνει τον πίνακα
                               στο πεδίο του χρόνου

fft_execute(fft_buff);       // Συνάρτηση μετασχηματισμού FFT που εκτελεί την συνάρτηση
                               FFT

fft_output(fft_buff, output); // Συνάρτηση μετασχηματισμού FFT που εξάγει το τελικό φάσμα
                               στον πίνακα output

for (i=0; i<=FFT_N/2; i++)   // Το i παίρνει τιμές από 0-31 αφού FFT_N/2=32
{
Serial.println(output[i]);    // Τυπώνουμε τις 32 τιμές μία μία τις τιμές χωρίς καθυστέρηση
}
delay(2000);                 // Τρέχει ο κώδικας και βγάζει τιμές κάθε 2 δευτ/πτα περίπου
}

```

Κ Ε Φ Α Λ Α Ι Ο 6

ΚΩΔΙΚΑΣ MATLAB

6.1 ΕΙΣΑΓΩΓΗ

Ο κώδικας που ακολουθεί είναι γραμμένος στο πρόγραμμα Matlab, όπου πλέον μπορούμε απεικονίσουμε γραφικά τα αποτελέσματα του φάσματος του σήματος καθώς μεταβάλλεται με την πάροδο του χρόνου.

6.2 ΚΩΔΙΚΑΣ

```
clear; % Αφαιρεί όλες τις μεταβλητές από το workspace
clc; % Καθαρίζει το command window
fs=8870; % Συχνότητα δειγματοληψίας που θα λειτουργήσει
ο Arduino
port=serial('COM3', 'BaudRate', 9600); % Ανοίγει η σειριακή θύρα COM3
fopen(port); % Ανοίγει η σειριακή θύρα COM3
x=linspace(0,fs/2,64); % Δημιουργεί μια σειρά διανυσμάτων από 0 έως 64
val=zeros(64,1); % Δημιουργεί έναν μηδενικό πίνακα με 64 στοιχεία
for i=1:64 % Βρόχος
val(i,1)=fscanf(port,'%f'); % Διαβάζει τις τιμές από την σειριακή θύρα
end % Τέλος βρόχου
figure(1); % Άνοιγμα παραθύρου γραφικών
stem(x,val); % Δημιουργία γραφήματος
xlabel('Frequency'); % Προσθήκη ετικέτας στον οριζόντιο άξονα
```

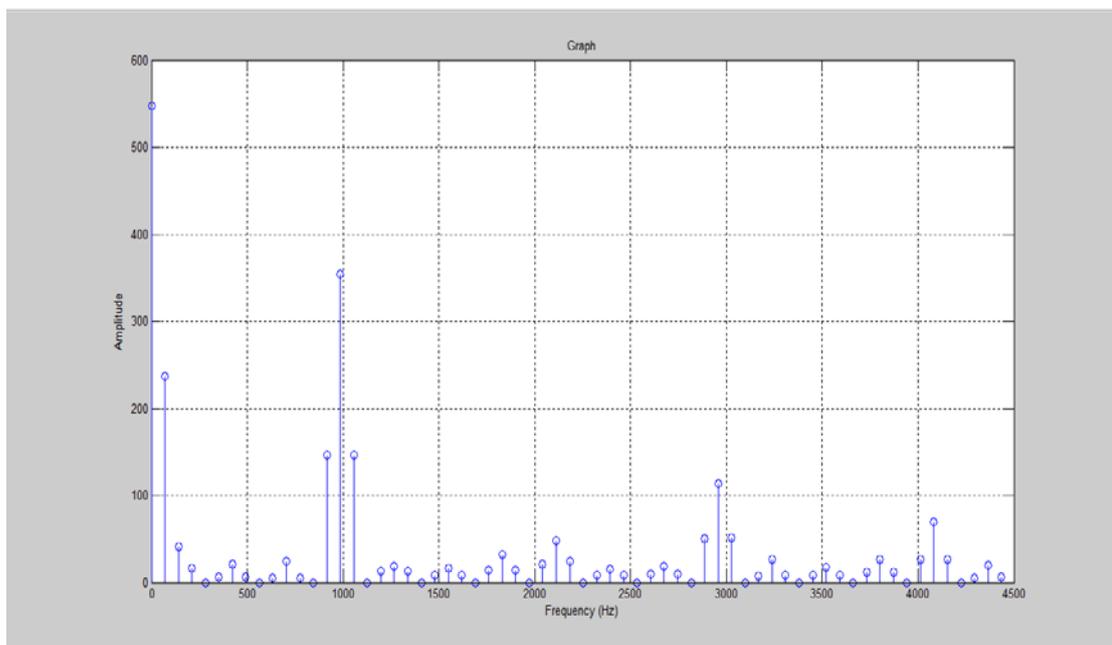
ylabel('Amplitude');	<i>% Προσθήκη ετικέτας στον κατακόρυφο άξονα</i>
title('Graph');	<i>% Τίτλος γραφήματος</i>
grid on;	<i>% Δημιουργία πλέγματος</i>
pause(0.1);	<i>% Μικρή καθυστέρηση για να μην κολλάει το πρόγραμμα όταν θα κάνει την επανάληψη</i>
fclose(port);	<i>% Κλείνει η σειριακή θύρα</i>
delete(port);	<i>% Διαγράφει την σειριακή θύρα</i>

6.3 ΣΥΜΠΕΡΑΣΜΑ - ΑΠΟΤΕΛΕΣΜΑ

Υλοποιώντας τον κώδικα στο Matlab και με συνδεδεμένη την πλακέτα Arduino στον υπολογιστή μας είμαστε πλέον σε θέση να βλέπουμε τη γραφική παράσταση του φάσματος ενός σήματος καθώς μεταβάλλεται με την πάροδο του χρόνου.

Το φάσμα που προκύπτει είναι διακριτό, και όχι συνεχές, επειδή η τετραγωνική μορφή είναι περιοδική .

Ορίσαμε στον κώδικά μας η θεμελιώδης συχνότητα του σήματος να είναι 1Khz, άρα στο φάσμα διακρίνονται συνιστώσες στις συχνότητες που είναι ακέραια – περιττά πολλαπλάσια αυτής της συχνότητας του σήματος δηλ. 1-3-5-7...KHz. Εμείς στην γραφική μας παράσταση βλέπουμε μέχρι 3KHz. Όσο μεγαλώνει η συχνότητα τόσο φθίνει το πλάτος.



*Γραφική παράσταση του φάσματος ενός σήματος καθώς μεταβάλλεται με την
πάροδο του χρόνου.*

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] http://cgi.di.uoa.gr/~k14/sig_sys1.pdf
- [2] http://anamorfosi.teicm.gr/ekp_yliko/e-notes/Data/digisignal/main.htm
- [3] <http://www.samos.aegean.gr/actuar/dlekkas/forecasting%20and%20sim/DSP-Chapter1.pdf>
- [4] <http://eclass.uoa.gr/modules/document/file.php>
- [5] <http://el.wikipedia.org/wiki>
- [6] http://www.eng.ucy.ac.cy/ece220/Lecture14_2010.pdf
- [7] Σύγγραμμα «ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΟΣ» του συγγραφέα MONSON H. HAYES
- [8] <http://el.wikipedia.org/wiki/Arduino>
- [9] <http://arduino.cc/en/Main/arduinoBoardUno>
- [10] <http://t-h.wikispaces.com/file>
- [11] <http://www.microplanet.gr/tutorials/microcontrollers/arduino>
- [12] <http://deltahacker.gr/2009/08/01/arduino-intro/>
- [13] http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf
- [14] <http://www2.ucy.ac.cy/~georgios/bookfiles/MATLABbook.pdf>
- [15] <http://users.sch.gr/sarellis/notes%20Matlab.htm>
- [16] <http://www2.ucy.ac.cy/~georgios/courses/mas191/Chapter5.pdf>
- [17] <http://edusoft.civil.auth.gr/TE4800/Matlab%20Notes%20and%20Codes/Matlab%20Notes.pdf>
- [18] Σύγγραμμα «ΣΗΜΑΤΑ & ΣΥΣΤΗΜΑΤΑ ΜΕ MATLAB» των Παλαμίδη Αλέξη και Βελώνη Αναστασίας
- [19] http://aetos.it.teithe.gr/~gouliana/files/neuronics/MM_ERGASTHRIO_Matlab_2011_kef_1_7.pdf
- [20] http://www2.ucy.ac.cy/~xenophon/pubs/matlab_intro.pdf
- [21] http://nereus.mech.ntua.gr/courses/control/control_pdf/matlab_primer.pdf