



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αναγνώριση κινήσεων (gestures) με χρήση
επιταχυνσιόμετρου και Arduino

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ Ν. ΡΩΜΑΝΙΑΔΗ

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αναγνώριση κινήσεων (gestures) με χρήση επιταχυνσιόμετρου και Arduino

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ Ν.ΡΩΜΑΝΙΑΔΗ

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ^η Οκτωβρίου 2014.

(Υπογραφή)

.....

Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

(Υπογραφή)

.....

Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π

(Υπογραφή)

.....

Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2014

(Υπογραφή)

.....

Ιωάννης Ν. Ρωμανιάδης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ιωάννης Ν. Ρωμανιάδης, 2014.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Σκοπός αυτής της διπλωματικής είναι η κατασκευή και παρουσίαση μίας συσκευής η οποία μπορεί να αναγνωρίζει χειρονομίες (gestures). Η συσκευή κατασκευάστηκε με τη χρήση της τεχνολογίας Arduino και ενός επιταχυνσιόμετρου ενώ για καλύτερα αποτελέσματα χρησιμοποιήθηκε και ένα γυροσκόπιο.

Το Arduino λαμβάνει από το επιταχυνσιόμετρο και το γυροσκόπιο τα δεδομένα και αφού γίνει η επεξεργασία τους μέσω ενός φίλτρου Kalman τότε υλοποιείται η αναγνώριση των χειρονομιών. Η εφαρμογή είναι γραμμένη σε Wiring μία παραλλαγή της C/C++ και σε Java ενώ η αναγνώριση γίνεται με την βοήθεια μίας παραλλαγής του αλγορίθμου \$1 Unistroke Recognizer. Η σύνδεση του Arduino με τον υπολογιστή γίνεται ασύρματα με την βοήθεια μίας συσκευής Bluetooth.

Στο θεωρητικό υπόβαθρο της εργασίας γίνεται εκτενής αναφορά στην έννοια της αναγνώρισης χειρονομιών στην αλληλεπίδραση ανθρώπου μηχανής καθώς και στις έννοιες του Internet of Things και στο Open Hardware. Επιπλέον παρουσιάζεται αναλυτικά ο τρόπος της υπολογιστικής υλοποίησης της όλης διαδικασίας.

Η μελέτη και η υλοποίηση αυτή είχε σαν αποτέλεσμα την αποπεράτωση μίας εφαρμογής που δέχεται χειρονομίες εξ αποστάσεως και είναι ένα σύγχρονο παράδειγμα αλληλεπίδρασης ανθρώπου μηχανής. Ο σχεδιασμός και η μεθοδολογία που ακολουθήθηκε μπορεί να γίνει οδηγός για την κατασκευή περισσότερων εφαρμογών που θα έδιναν μία άλλη διάσταση στη χρήση των ηλεκτρονικών συσκευών και θα είχαν εφαρμογή σε πολλούς τομείς της αλληλεπίδρασης ανθρώπου μηχανής.

Λέξεις Κλειδιά

Αλληλεπίδραση ανθρώπου μηχανής, αναγνώριση χειρονομιών, Internet of things, Open Hardware, Arduino, επιταχυνσιόμετρο, γυροσκόπιο, Kalman Filter.

Abstract

The purpose of this project is the construction and presentation of a device that can recognize gestures. The device is built using Arduino technology and an accelerometer. For better results a gyroscope is combined.

The Arduino receives from the accelerometer and the gyroscope data and a Kalman filter processes them. After that the recognition of the gestures takes place. The application is written in Wiring a variant of C/C++ and Java. The identification is done with the help of a variant of the algorithm \$1 Unistroke Recognizer. The connection of the Arduino to the computer is done wirelessly with the aid of a Bluetooth.

The theoretical background of this thesis includes information to the concept of gesture recognition in human machine interaction, to the concept of Internet of Things and to Open Hardware. Additionally is shown in detail the computational implementation of the entire process.

The design and implementation of this project resulted in the completion of an application that accepts remote gestures and is a modern example of man-machine interaction. The design and methodology that was followed can become guide to the construction of more applications that would give another dimension to the use of electronic devices and would have application in many fields of human machine interaction.

Keywords

Human machine interaction, gesture recognition, Internet of things, Open Hardware, Arduino, accelerometer, gyroscope, Kalman Filter.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Τεχνολογίας Πληροφορικής & Υπολογιστών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π, στα πλαίσια των ερευνητικών δραστηριοτήτων γύρω από την αλληλεπίδραση ανθρώπου μηχανής.

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Κόλλια Στέφανο για την ανάθεση της διπλωματικής και τον ερευνητή κ. Καρπούζη Κωνσταντίνο για την καθοδήγηση, την επίβλεψη και την υποστήριξη με πληροφορίες, συμβουλές και υποδείξεις καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας.

Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερα την οικογένειά μου, τους φίλους μου και όσους ανθρώπους ήταν κοντά μου όλο αυτό το διάστημα.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	17
1.1 Αναγνώριση χειρονομιών στην αλληλεπίδραση.....	17
1.2 Αντικείμενο διπλωματικής.....	17
1.2.1 Συνεισφορά εργασίας.....	18
1.3 Διάρθρωση εργασίας.....	18
ΚΕΦΑΛΑΙΟ 2: Θεωρητικό υπόβαθρο.....	21
2.1 Αλληλεπίδραση ανθρώπου υπολογιστή.....	21
2.2 Επικοινωνία ανθρώπου μηχανής με χρήση χειρονομιών.....	23
2.2.1 Εφαρμογές.....	24
2.2.2 Εργαλεία αναγνώρισης χειρονομιών.....	25
2.3 Internet of Things (IoT).....	26
2.4 Open Hardware.....	29
ΚΕΦΑΛΑΙΟ 3: Υλικό μέρος (Hardware).....	31
3.1 Arduino.....	31
3.2 Arduino Uno.....	32
3.2.1 Τροφοδοσία.....	33
3.2.2 Ακροδέκτες.....	34
3.2.3 Μνήμη.....	36
3.3 Accelerometer.....	36
3.3.1 Εφαρμογές.....	37

3.3.2 Λειτουργία.....	37
3.4 Γυροσκόπιο.....	38
3.5 MPU 6050.....	39
3.5.1 I2C bus.....	40
3.6 Bluetooth.....	42
3.6.1 Bluetooth mate.....	43
3.7 Περιγραφή συνδέσεων.....	44
ΚΕΦΑΛΑΙΟ 4: Λογισμικό μέρος (Software).....	47
4.1 Arduino Ide.....	47
4.2 Θόρυβος και φίλτρα.....	51
4.2.1 Kalman Filter.....	52
4.2.2 Μαθηματική ανάλυση Kalman filter.....	53
4.3 Processing.....	59
4.4 1\$ Unistroke αλγόριθμος αναγνώρισης.....	60
ΚΕΦΑΛΑΙΟ 5: Υλοποίηση.....	67
5.1 Καταγραφή δεδομένων.....	67
5.2 Φιλτράρισμα.....	68
5.2.1 Complementary Filter.....	68
5.2.2 Kalman Filter.....	69
5.2.3 Επιλογή δεδομένων.....	73
5.3 Αναγνώριση χειρονομιών.....	75

ΚΕΦΑΛΑΙΟ 6: Επίλογος.....	81
6.1 Συμπεράσματα.....	81
6.2 Μελλοντικές επεκτάσεις.....	81
Βιβλιογραφία.....	83

Πίνακας Εικόνων

Εικόνα 1: Πεδία αλληλεπίδρασης ανθρώπου-υπολογιστή.....	22
Εικόνα 2: Αναγνώριση χειρονομιών στην αλληλοεπίδραση.....	23
Εικόνα 3: Αναγνώριση νοηματικής γλώσσα.....	24
Εικόνα 4: Χρήση ειδικών γαντιών για αναγνώριση χειρονομιών.....	25
Εικόνα 5: Παράδειγμα χρήσης κάμερας βάθους.....	26
Εικόνα 6: Παραδείγματα χρήσης IoT.....	28
Εικόνα 7: Open hardware.....	29
Εικόνα 8: Βασικά χαρακτηριστικά Arduino Uno.....	32
Εικόνα 9: Arduino Uno.....	32
Εικόνα 10: Τροφοδοσία μέσω μετασχηματιστή.....	33
Εικόνα 11: Τροφοδοσία μέσω μπαταρίας.....	33
Εικόνα 12: Ακροδέκτες τροφοδοσίας.....	34
Εικόνα 13: Ψηφιακοί ακροδέκτες.....	35
Εικόνα 14: Ακροδέκτες Arduino Uno.....	37
Εικόνα 15: Λειτουργία επιταχυνσιόμετρου.....	37
Εικόνα 16: MPU 6050.....	39
Εικόνα 17: Στοιχεία I2C bus.....	40
Εικόνα 18: Ακολουθία έναρξης και τερματισμού.....	40
Εικόνα 19: Δεδομένα στον I2C bus.....	41
Εικόνα 20: Διευθύνσεις I2C bus.....	41
Εικόνα 21: Διαδικασίες write και read.....	42
Εικόνα 22: Bluetooth.....	43

Εικόνα 23: Bluetooth mate.....	44
Εικόνα 24: Σύνδεση Arduino-Mpu.....	45
Εικόνα 25: Σύνδεση Arduino-Bluetooth.....	45
Εικόνα 26: Εγκατασταση Bluetooth.....	46
Εικόνα 27: Εύρεση COM.....	46
Εικόνα 28: Βασικές λειτουργίες Arduino IDE.....	47
Εικόνα 30: Επιλογή σωστής πλακέτας.....	51
Εικόνα 31: Processing.....	59
Εικόνα 32: Διαφορετικός αριθμών σημείων για το ίδιο σχήμα.....	60
Εικόνα 33: Επαναδειγματοληψία.....	61
Εικόνα 34: Περιστροφή.....	63
Εικόνα 35: Τιμές FS_SEL.....	68
Εικόνα 36: Complementary Filter.....	69
Εικόνα 37: Δεδομένα Arduino αρχικής στιγμής.....	73
Εικόνα 38: Δεδομένα Arduino τυχαίας στιγμής.....	74
Εικόνα 39: Γραφική παράσταση με όλους τους τρόπους συλλογής δεδομένων...	75
Εικόνα 40: Σχήματα αναγνώρισης.....	76
Εικόνα 41: Αναγνώριση circle.....	77
Εικόνα 41: Αναγνώριση v.....	77
Εικόνα 42: Αναγνώριση pigtail.....	78
Εικόνα 43: Αναγνώριση caret.....	78

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Αναγνώριση χειρονομιών στην αλληλεπίδραση

Η αλληλεπίδραση ανθρώπου υπολογιστή ορίζεται ως το επιστημονικό πεδίο του οποίου η μελέτη αφορά την αλληλεπίδραση και επικοινωνία μεταξύ ανθρώπων και υπολογιστικών μηχανών. Η αλληλεπίδραση αυτή γίνεται στο επίπεδο της διεπαφής χρήστη, μέσω κατάλληλου λογισμικού και υλικού. Καθημερινά κάθε άνθρωπος έρχεται σε επαφή με πολλές ηλεκτρονικές συσκευές ενώ την τελευταία δεκαετία υπάρχει αύξηση της κυκλοφορίας τέτοιων νέων συσκευών. Η τεχνολογία αυτή πρέπει να είναι προσιτή και κατανοητή στον μέσο χρήστη όπως είναι και η επικοινωνία μεταξύ ανθρώπων. Η επικοινωνία μεταξύ ανθρώπων γίνεται με πολλούς τρόπους όπως για παράδειγμα τον λόγο, τις χειρονομίες, τη στάση του σώματος, εκφράσεις προσώπου κ.α. Ο συνδυασμός όλων αυτών παράγει πλουσιότερη ποικιλία έκφρασης.

Η κάμερα και το μικρόφωνο, δύο πολύ διαδεδομένες συσκευές, μπορούν να ικανοποιήσουν κάποιους από τους φυσικούς τρόπους έκφρασης των ανθρώπων που αναφέρθηκαν παραπάνω, όπως τον λόγο και την έκφραση του προσώπου. Για τους υπόλοιπους τρόπους έκφρασης όμως, αναζητούνται νέοι τρόποι αλληλεπίδρασης για μεγαλύτερη διευκόλυνση. Στα πλαίσια αυτής της αναζήτησης, η χρήση χειρονομιών αποτελεί έναν φυσικό, πρωτοποριακό και μοντέρνο τρόπο για μη λεκτική επικοινωνία που βρίσκει εφαρμογή σε πολλές εφαρμογές υπολογιστικής αλληλεπίδρασης αλλά και στη νοηματική γλώσσα.

Για τον εντοπισμό της κίνησης του ανθρώπινου σώματος χρειαζόμαστε πιο εξειδικευμένο εξοπλισμό. Έτσι λοιπόν, γίνονται συνεχείς έρευνες και μελέτες, ώστε να τελειοποιηθούν ψηφιακοί αισθητήρες με μεγάλη ακρίβεια και αποτελεσματικότητα στην αντίληψη των μηχανών με ψηφιακή επεξεργασία πολυμέσων ή με εκπομπή διάφορων ακτινών ή διάφορων αισθητήρων για την κατανόηση του φυσικού χώρου από τους ηλεκτρονικούς υπολογιστές. Ένας τέτοιος συνδυασμός από πρωτοποριακές συσκευές σε συνεργασία με την ανάπτυξη κατάλληλων λογισμικών θα έδινε μία νέα πνοή στον τομέα της αλληλεπίδρασης ανθρώπου μηχανής.

1.2 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη και κατασκευή μίας συσκευής αναγνώρισης χειρονομιών με την βοήθεια της τεχνολογίας Arduino. Το πρόγραμμα για την αναγνώριση των χειρονομιών είναι γραμμένο σε γλώσσα Wiring μία παραλλαγή της C/C++ και σε Java. Η επικοινωνία της συσκευής με κάποιον υπολογιστή γίνεται ασύρματα με την βοήθεια ενός Bluetooth.

Τα δεδομένα τα οποία είναι απαραίτητα για την αναγνώριση συλλέγονται με την βοήθεια ενός επιταχυνσιόμετρου και ενός γυροσκοπίου. Το πρόβλημα κατά τη διάρκεια της λήψης των

δεδομένων είναι ο θόρυβος που παρουσιάζεται. Για αυτό το λόγο μετά την συλλογή τους ακολουθεί η επεξεργασία τους με την βοήθεια ενός Kalman Filter.

Ο αλγόριθμος για την αναγνώριση βασίζεται σε μία παραλλαγή του \$1 Unistroke Recognizer από το πανεπιστήμιο της Washington.

1.2.1 Συνεισφορά εργασίας

- Παρουσιάστηκε λεπτομερώς το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση και την θεωρητική θεμελίωση της εφαρμογής.
- Το πρόγραμμα υλοποιήθηκε με τέτοιο τρόπο ώστε να παρέχει μία ευέλικτη εφαρμογή με χαμηλό υπολογιστικό κόστος. Η συνεισφορά της υπολογιστικής εφαρμογής περιλαμβάνει τα παρακάτω:
 - Έγινε μελέτη της λειτουργίας του επιταχυνσιόμετρου και παρουσιάστηκαν οι λόγοι για την χρήση του γυροσκοπίου.
 - Παρουσιάστηκαν δύο φίλτρα για την μείωση θορύβου και για ακριβέστερα αποτελέσματα.
 - Εξετάστηκε ο αλγόριθμος αναγνώρισης χειρονομιών.
- Αξιολογήθηκε η μέθοδος. Διερευνήθηκαν τα πλεονεκτήματα και τα δυνατά σημεία της μεθόδου καθώς και οι πρακτικοί περιορισμοί και οι αδυναμίες της υλοποίησης.
- Έγιναν προτάσεις για περαιτέρω τροποποίηση και επέκταση του προγράμματος.

1.3 Διάρθρωση εργασίας

- Το πρώτο κεφάλαιο της εργασίας περιέχει τα εισαγωγικά στοιχεία που είναι απαραίτητα για να κατανοήσει ο αναγνώστης το αντικείμενο και το πεδίο εφαρμογής της διπλωματικής.
- Το δεύτερο κεφάλαιο εστιάζει σε βασικές θεωρητικές έννοιες στις οποίες βασίζεται η παρούσα διπλωματική όπως είναι η αλληλεπίδραση ανθρώπου μηχανής και η αναγνώριση χειρονομιών. Επίσης γίνεται αναφορά σε δύο έννοιες, τις Internet of Things και Open Hardware, οι οποίες συνδέονται με την εφαρμογή με άμεσο τρόπο.

- Στο τρίτο κεφάλαιο παρουσιάζεται το υλικό μέρος της εφαρμογής. Περιγράφεται η βασική τεχνολογία που χρησιμοποιήθηκε δηλαδή το Arduino. Δίνονται τα γενικά χαρακτηριστικά της τεχνολογίας και περιγράφονται λεπτομερώς τα χαρακτηριστικά του Arduino Uno δηλαδή του μοντέλου που χρησιμοποιήθηκε. Στη συνέχεια περιγράφονται οι λειτουργίες των επιταχυνσιόμετρων και των γυροσκοπίων και τα χαρακτηριστικά του MPU 6050. Τέλος γίνεται αναφορά στην τεχνολογία Bluetooth και περιγράφεται με αναλυτικά ο τρόπος σύνδεσης όλων των συσκευών.
- Στο τέταρτο κεφάλαιο παρουσιάζεται το λογισμικό και οι αλγόριθμοι που χρησιμοποιήθηκαν. Συγκεκριμένα περιγράφεται το Arduino IDE, το λογισμικό για την σχεδίαση προγραμμάτων στο Arduino. Γίνεται αναφορά στον θόρυβο και στα φίλτρα ενώ περιγράφεται αναλυτικά το φίλτρο Kalman. Στη συνέχεια παρουσιάζεται το Processing ένα σχεδιαστικό περιβάλλον που χρησιμοποιήθηκε για να οπτικοποιήσει τα αποτελέσματα καθώς και ο αλγόριθμος για την αναγνώριση των χειρονομιών.
- Το πέμπτο κεφάλαιο είναι το κεφάλαιο της υλοποίησης. Παρουσιάζονται όλα τα στάδια για την λειτουργία της συσκευής όπως η συλλογή δεδομένων, το φιλτράρισμα, η επικοινωνία με τον υπολογιστή καθώς και ο αναλυτικός τρόπος για την αναγνώριση των χειρονομιών.
- Στο έκτο κεφάλαιο παρουσιάζονται τα συμπεράσματα από την υλοποίηση καθώς και οι μελλοντικές επεκτάσεις της εφαρμογής.

ΚΕΦΑΛΑΙΟ 2

2.1 Αλληλεπίδραση ανθρώπου υπολογιστή

Η αλληλεπίδραση ανθρώπου υπολογιστή (HCI) είναι η επιστήμη της πληροφορικής που ασχολείται με το σχεδιασμό, την ανάπτυξη και την αξιολόγηση διαδραστικών υπολογιστικών συστημάτων που προορίζονται για ανθρώπινη χρήση και με τη μελέτη σημαντικών φαινομένων γύρω από αυτά. Η επιστήμη της αλληλεπίδρασης αποτελείται από τρία μέρη τα οποία είναι ο άνθρωπος, ο υπολογιστής και ο τρόπος που αυτά αλληλεπιδρούν μεταξύ τους. Η έννοια άνθρωπος περιλαμβάνει τόσο έναν μεμονωμένο χρήστη όσο και μία ομάδα χρηστών που εργάζονται μαζί. Ο όρος υπολογιστής αναφέρεται σε οποιαδήποτε τεχνολογία όπως είναι τα κινητά τηλέφωνα, οι προσωπικοί υπολογιστές, οι εφαρμογές διαδικτύου και τα μεγάλα υπολογιστικά συστήματα. Ο όρος αλληλεπίδραση αναφέρεται στην οποιαδήποτε επικοινωνία μεταξύ των δύο, μέσω μίας διεπιφάνειας. Η αλληλεπίδραση ανθρώπου υπολογιστή μερικές φορές αναφέρεται ως αλληλεπίδραση ανθρώπου μηχανής (MMI). Συχνά θεωρείται ως η τομή της επιστήμης των υπολογιστών, επιστήμες της συμπεριφοράς, το σχεδιασμό και πολλούς άλλους τομείς σπουδών.

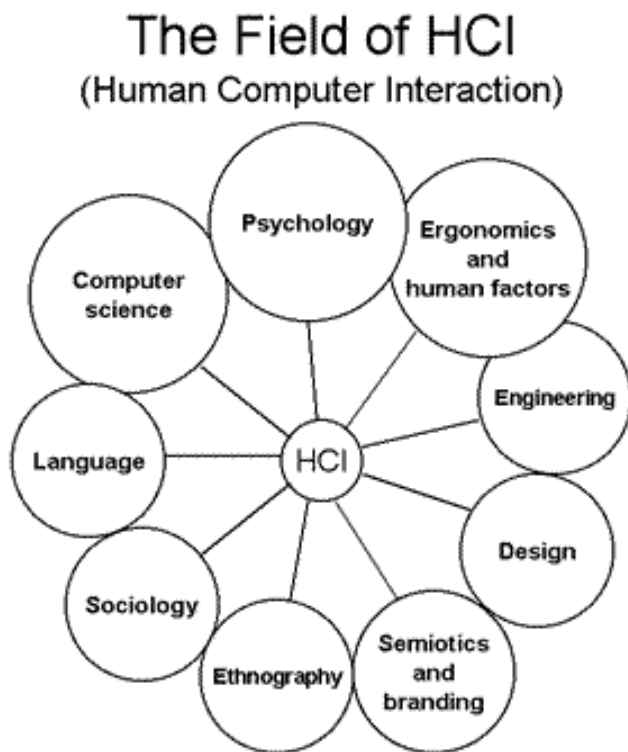
Η αλληλεπίδραση ανθρώπου υπολογιστή μελετά και τον άνθρωπο και την μηχανή για αυτό και αντλεί πληροφορίες και από τις δύο πλευρές. Από την πλευρά της μηχανής, οι τεχνικές σε γραφικά ηλεκτρονικών υπολογιστών, τα λειτουργικά συστήματα, οι γλώσσες προγραμματισμού και τα περιβάλλοντα ανάπτυξης είναι σημαντικά. Από την ανθρώπινη πλευρά, η θεωρία της επικοινωνίας, η γλωσσολογία, οι κοινωνικές επιστήμες, η γνωστική ψυχολογία, και οι άνθρωποι παράγοντες, όπως η ικανοποίηση των χρηστών ηλεκτρονικών υπολογιστών είναι σχετικές.

Η επιστήμη της αλληλεπίδρασης ανθρώπου υπολογιστή διαφέρει σε σχέση με τους άλλους κλάδους της πληροφορικής, καθώς είναι υποχρεωμένη να μελετήσει πέρα των τεχνολογιών και των μεθοδολογιών σχεδίασης και λειτουργίας υπολογιστικών συστημάτων, τις συμπεριφορές των χρηστών που θα αλληλεπιδράσουν με τα υπολογιστικά αυτά συστήματα είτε ως μονάδες είτε ως μέλη μίας ομάδας. Λόγω της ιδιαιτερότητας αυτής υπάρχουν πάρα πολλοί επιστημονικοί τομείς που συνεισφέρουν στην επιστήμη της αλληλεπίδρασης ανθρώπου υπολογιστή, από τους οποίους αντλεί συμπεράσματα.

Για την σχεδίαση ενός διαδραστικού συστήματος απαιτούνται γνώσεις από πολλά επιστημονικά πεδία: την ψυχολογία και τη γνωστική επιστήμη (γνώση για την αντίληψη του χρήστη, την γνωστική του ικανότητα και την ικανότητα επίλυσης προβλημάτων), την εργονομία (γνώση για τις φυσικές ικανότητες του χρήστη), την κοινωνιολογία (κατανόηση του ευρύτερου πλαισίου μέσα στο οποίο λαμβάνει χώρα η διάδραση), την επιστήμη των υπολογιστών (ανάπτυξη της απαιτούμενης τεχνολογίας), τη σχεδίαση γραφικών (για την δημιουργία μίας αποτελεσματικής

παρουσίασης της διεπιφάνειας), την συγγραφή τεχνικών κειμένων (για την σύνταξη των συνοδευτικών εγχειριδίων του συστήματος) κ.α.

Για το λόγο αυτό, παρόλο που ο όρος επικοινωνία ανθρώπου μηχανής περιλαμβάνει όλα τα ανωτέρω μέρη, στην πράξη υπάρχουν εξειδικευμένοι επιστήμονες που επικεντρώνουν το ενδιαφέρον τους σε καθένα από τους παραπάνω τομείς. Έτσι, το τελικό σύστημα, αποτελεί το προϊόν της συνεργασίας εξειδικευμένων επιστημόνων όλων των παραπάνω κλάδων.



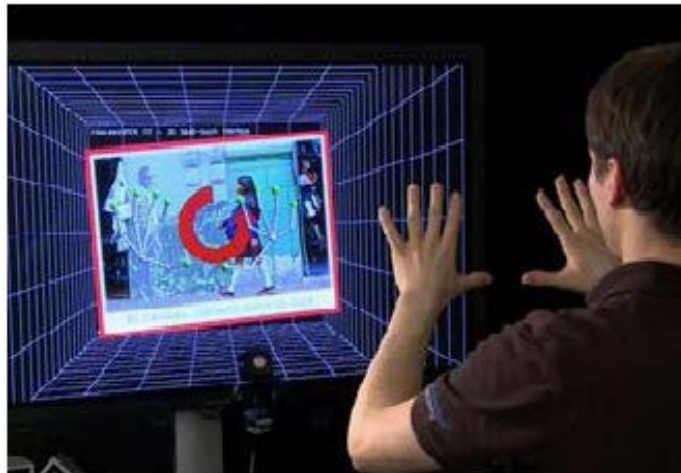
Εικόνα 1: Πεδία αλληλεπίδρασης ανθρώπου-υπολογιστή

Βασικός στόχος είναι η βελτίωση της αλληλεπίδρασης μεταξύ των ανθρώπων και των υπολογιστών κάνοντας τους υπολογιστές πιο εύχρηστους και προσιτούς στις ανάγκες του χρήστη. Συγκεκριμένα η αλληλεπίδραση ανθρώπου μηχανής ασχολείται με την ανάπτυξη μεθοδολογιών και διαδικασιών για τον σχεδιασμό διασυνδέσεων καθώς και με την υλοποίηση των διασυνδέσεων όπως είναι οι αποδοτικοί αλγόριθμοι και τα διάφορα εργαλεία λογισμικού. Αναζητούνται νέοι τύποι διασύνδεσης και τεχνικές αλληλεπίδρασης πιο προσιτές στον άνθρωπο. Ένας μακροπρόθεσμος στόχος της αλληλεπίδρασης ανθρώπου υπολογιστή είναι ο σχεδιασμός συστημάτων που ελαχιστοποιούν το φράγμα μεταξύ του γνωστικού μοντέλου του ανθρώπου για το τι θέλει να πετύχει και την κατανόηση του υπολογιστή της εργασίας του χρήστη.

2.2 Επικοινωνία ανθρώπου μηχανής με χρήση χειρονομιών

Τα τελευταία χρόνια υπάρχει μεγάλο ενδιαφέρον για την βελτίωση της αλληλεπίδρασης μεταξύ ανθρώπου και υπολογιστή τόσο σε επίπεδο διεπαφής όσο και σε επίπεδο απόκρισης του συστήματος. Τα πιο διαδεδομένα μέσα για την αλληλεπίδραση είναι το ποντίκι και το πληκτρολόγιο ενώ με την χρήση καμερών και μικροφώνων δίνεται η δυνατότητα εισόδου και επεξεργασίας οπτικής και ηχητικής πληροφορίας στον υπολογιστή. Η βελτίωση του τομέα στοχεύει στην προσομοίωση των επικοινωνιακών δεξιοτήτων ενός ανθρώπου. Τα τελευταία χρόνια η επικοινωνία ανθρώπου υπολογιστή επικεντρώνεται στους τομείς που περιλαμβάνουν την αναγνώριση συναισθημάτων από το πρόσωπο και στην αναγνώριση χειρονομιών.

Η αναγνώριση χειρονομιών είναι ένας τομέας στην επιστήμη των υπολογιστών και της τεχνολογίας γλώσσας που έχει στόχο την ερμηνεία ανθρώπινων χειρονομιών μέσω μαθηματικών αλγορίθμων. Αυτές οι χειρονομίες μπορεί να προέρχονται από οποιαδήποτε σωματική κίνηση ή κατάσταση αλλά συνήθως προέρχονται από το πρόσωπο ή το χέρι. Πολλές προσεγγίσεις έχουν γίνει χρησιμοποιώντας κάμερες και αλγορίθμους υπολογιστικής όρασης με σκοπό την κατανόηση και ερμηνεία της νοηματικής γλώσσας. Ωστόσο, ο εντοπισμός και η αναγνώριση της στάσης του σώματος, το βάδισμα, και οι ανθρώπινες συμπεριφορές είναι, επίσης, το θέμα των τεχνικών αναγνώρισης χειρονομιών.



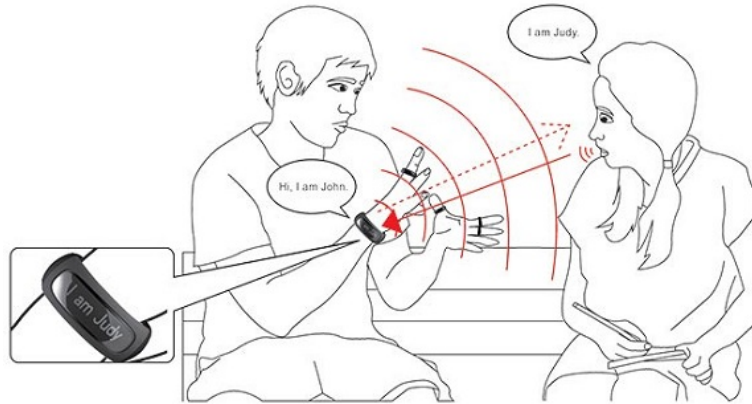
Εικόνα 2: Αναγνώριση χειρονομιών στην αλληλεπίδραση

Η αναγνώριση των χειρονομιών μπορεί να θεωρηθεί ως ένας τρόπος από τους υπολογιστές για να αρχίσουν να αντιλαμβάνονται και να καταλαβαίνουν την γλώσσα του σώματος ώστε να υπάρχει μία καλύτερη επικοινωνία μεταξύ ανθρώπων και υπολογιστών και να βοηθήσει στην κατασκευή νέων μέσων που θα καταστήσουν συμβατικές συσκευές εισόδου, όπως ποντίκια, πληκτρολόγια, ακόμα και οθόνες αφής.

2.2.1 Εφαρμογές

Η αναγνώριση χειρονομιών βρίσκει εφαρμογή σε πολλούς τομείς κάποιου από τους οποίους αναφέρονται παρακάτω:

- **Αναγνώριση νοηματικής γλώσσας.** Σε αρκετές εφαρμογές υπάρχει η δυνατότητα μετατροπής της ομιλίας σε κείμενο. Αντίστοιχα υπάρχει η δυνατότητα, από λογισμικά, της μετατροπής των συμβόλων της νοηματικής γλώσσας σε κείμενο ή σε ομιλία.



Εικόνα 3: Αναγνώριση νοηματικής γλώσσας

- **Ρομποτική ιατρική.** Με χρήση αισθητήρων επάνω στο σώμα του ασθενή και από την ανάγνωση των δεδομένων τους, μπορεί να γίνει η αποκατάσταση των ασθενών όπως είναι η περίπτωση του εγκεφαλικού.
- **Ένδειξη κατεύθυνσης μέσω της κατάδειξης.** Δηλαδή μέσω της καταγραφής της κίνησης και της αναγνώρισης του είδους της χειρονομίας μπορεί να προσδιοριστούν δηλώσεις ή οδηγίες για την συμπεριφορά κάποιου αντικείμενου. Η εφαρμογή αυτή είναι χρήσιμη στον τομέα της ρομποτικής.
- **Έλεγχος μέσω χειρονομιών προσώπου.** Η αναγνώριση χειρονομιών του προσώπου κάνει πιο εύκολο τον έλεγχο ενός υπολογιστή σε άτομα που αντιμετωπίζουν δυσκολία στην χρήση άλλων συσκευών όπως είναι το ποντίκι και το πληκτρολόγιο. Ένα τέτοιο παράδειγμα είναι η παρακολούθηση των ματιών για τον έλεγχο της κίνησης του δρομέα.
- **Εντυπωσιακή τεχνολογία παιχνιδιών.** Η αναγνώριση χειρονομιών μπορεί να χρησιμοποιεί στην αλληλεπίδραση με παιχνίδια ώστε η εμπειρία του χρήστη να γίνει πιο διαδραστική, ζωντανή και ρεαλιστική.

- **Εικονικοί ελεγκτές.** Σε πολλά συστήματα υπάρχει η δυνατότητα αντικατάστασης του φυσικού ελεγκτή με ένα σύστημα αναγνώρισης χειρονομιών. Τέτοια παραδείγματα είναι ο έλεγχος μίας τηλεόρασης ή ο έλεγχος λειτουργιών σε ένα αυτοκίνητο.

2.2.2 Εργαλεία αναγνώρισης χειρονομιών

Η αναγνώριση κινήσεων και χειρονομιών μπορεί να γίνει με πολλούς τρόπους και αρκετά εργαλεία κάποια από τα οποία είναι διαδεδομένα σε μεγάλο βαθμό στην καθημερινότητα ενώ κάποια άλλα βρίσκουν εφαρμογές σε ειδικές περιπτώσεις.

Controller-based χειρονομίες: Τέτοιοι ελεγκτές λειτουργούν ως προέκταση του σώματος. Όταν εκτελούνται οι χειρονομίες το λογισμικό μπορεί και τις αντιλαμβάνεται. Τέτοιο παράδειγμα είναι οι κινήσεις του ποντικού, οι οποίες σχετίζονται με την διαδρομή που πραγματοποιείται από το χέρι του χρήστη, όπως και το Wii Remote το οποίο με τις αλλαγές τις επιτάχυνσης καταλαβαίνει κινήσεις. Με παρόμοιο τρόπο λειτουργεί και η συσκευή αυτής της εργασίας. Ένα άλλο παράδειγμα είναι τα Audio Cubes τα οποία μπορούν να ανιχνεύσουν τις κινήσεις από τα χέρια και τα δάχτυλα και να βοηθήσουν στην σύνθεση μουσικής και ήχου.

Χρήση γαντιών: Τα ενσύρματα γάντια μπορούν να συνεισφέρουν στην όραση του υπολογιστή για την αναγνώριση της θέσης και της περιστροφής των χεριών, με τη χρήση μαγνητικών ή αδρανειακών συσκευών εντοπισμού. Επιπλέον, ορισμένα ενσύρματα γάντια μπορούν να ανιχνεύσουν την κάμψη των δαχτύλων με ένα υψηλό βαθμό ακρίβειας ή ακόμη και να παρέχουν απτική ανάδραση στο χρήστη, η οποία είναι μία προσομοίωση της αίσθησης της αφής. Το πρώτο εμπορικά διαθέσιμο χέρι-εντοπισμού, τύπου γάντι, ήταν η συσκευή DataGlove, ένα γάντι που μπορούσε να ανιχνεύσει τη θέση του χεριού, την κυκλοφορία και την κάμψη των δαχτύλων. Αυτό χρησιμοποιούσε καλώδια οπτικών ινών για να καλύπτει το πίσω μέρος του χεριού. Οι παλμοί φωτός που δημιουργούνται όταν τα δάχτυλα είναι λυγισμένα, διαρρέουν μέσα από μικρές ρωγμές και δημιουργείται μία μορφή απώλειας, δίνοντας έτσι μία προσέγγιση του σχήματος του χεριού.



Εικόνα 4: Χρήση ειδικών γαντιών για αναγνώριση χειρονομιών

Απλή κάμερα: Μία απλή 2D κάμερα μπορεί να αναγνωρίσει κινήσεις και χειρονομίες. Μία κανονική 2D κάμερα μπορεί να χρησιμοποιηθεί για την αναγνώριση χειρονομιών. Στην software based τεχνολογία η αναγνώριση χειρονομιών βασισμένη στο λογισμικό χρησιμοποιεί μία απλή κάμερα που μπορεί να ανιχνεύσει χειρονομίες, κινήσεις των χεριών, καθώς και τα δάκτυλα σε υψηλή ακρίβεια. Αυτά έχουν ήδη ενσωματωθεί σε ultrabooks Yoga της Lenovo, Vega Pantech του LTE smartphones, Smart τηλεοράσεις και σε άλλες συσκευές.

Κάμερα Βάθους: Με την χρήση εξειδικευμένων φωτογραφικών μηχανών υπάρχει η δυνατότητα να δημιουργηθεί ένας χάρτης βάθους για το τι μπορεί κανείς να δει μέσα από την κάμερα σε μικρή κλίμακα, και να χρησιμοποιήσει αυτά τα δεδομένα για την προσέγγιση μίας 3D αναπαράστασης του ορατού αντικειμένου. Αυτό είναι αποτελεσματικό για την ανίχνευση των χειρονομιών λόγω της μικρής κλίμακας του βάθους τους. Ένα παράδειγμα της κατηγορίας αυτής αποτελεί το Kinect. Πρόκειται για μία συσκευή ανίχνευσης κίνησης, από τη Microsoft, για την βίντεο-κονσόλα παιχνιδιών Xbox 360 και τα Windows PCs. Το Kinect, χρησιμοποιώντας παράλληλα πληροφορία βάθους, επιτρέπει στους χρήστες να ελέγχουν και να αλληλεπιδρούν με το Xbox 360, χωρίς να χρειάζεται να αγγίξουν ένα χειριστήριο παιχνιδιών, μέσα από μία φυσική διεπαφή του χρήστη, χρησιμοποιώντας χειρονομίες.



Εικόνα 5: Παράδειγμα χρήσης κάμερας βάθους

Stereo κάμερες: Χρησιμοποιώντας δύο κάμερες των οποίων οι σχέσεις μεταξύ τους είναι γνωστές, μία 3D αναπαράσταση μπορεί να προσεγγιστεί από την έξοδο τους.

2.3 Internet of Things (IoT)

Η γρήγορη επέκταση του διαδικτύου καθώς και η συνεχής σμίκρυνση των ηλεκτρονικών συσκευών οδηγούν σε μία ολοένα αυξανόμενη τάση της διασύνδεσης τέτοιων ποικίλων συσκευών ή εξαρτημάτων με το διαδίκτυο. Έξυπνα κινητά τηλέφωνα, ηλεκτρονικά βιβλία, αυτοκίνητα, ηλεκτρικές συσκευές και ηλεκτρονικές ετικέτες προϊόντος έχουν την δυνατότητα να συνδεθούν στο διαδίκτυο και να αλληλεπιδρούν με ανθρώπους και με υπολογιστικά συστήματα.

Η διασύνδεση αυτή πραγματοποιείται για να ενισχύσει τη λειτουργικότητα και τον έλεγχο των διασυνδεδεμένων συσκευών, προσδίδοντας νέα χαρακτηριστικά ή τρόπους χρήσης που απουσίαζαν από τον αρχικό σχεδιασμό της συσκευής. Μάλιστα, η τάση αυτή είναι τόσο έντονη που έχει οδηγήσει στη γέννηση του όρου “Διαδίκτυο Αντικειμένων” (Internet of Things), κατά τον οποίο κάθε συσκευή έχει μία μοναδική ταυτότητα και μπορεί να ελεγχθεί ή να ερωτηθεί για την κατάστασή της μέσω του διαδικτύου.

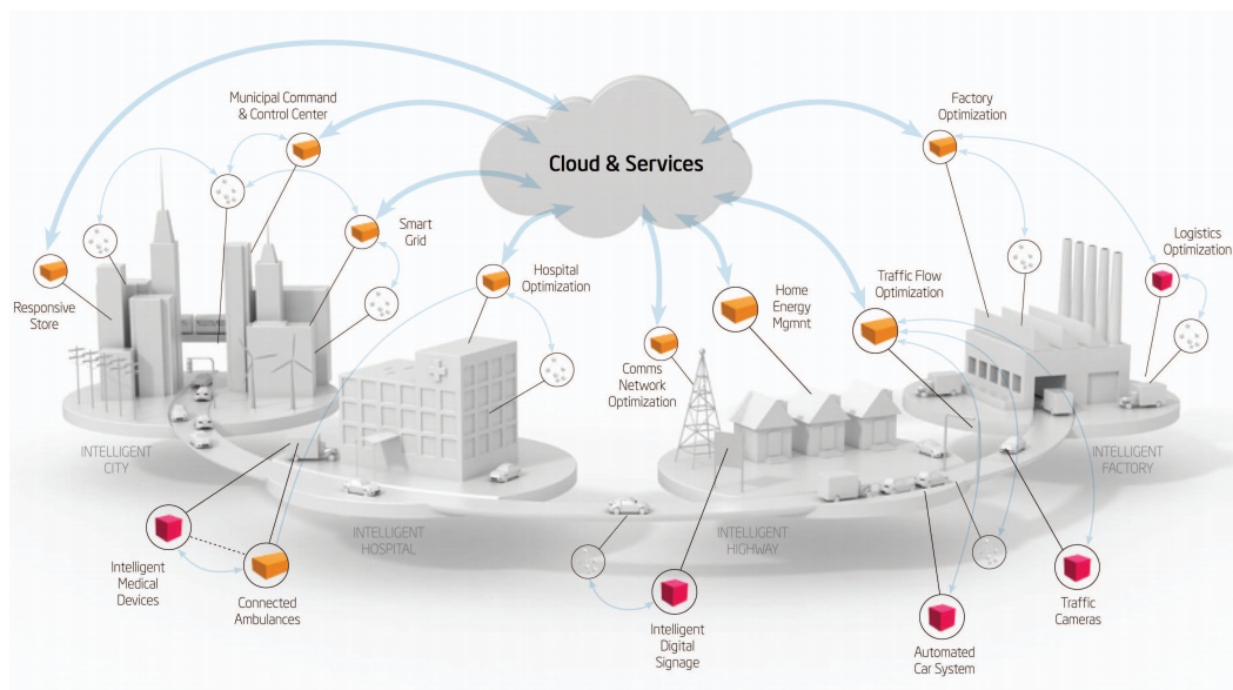
Το IoT είναι η νέα επανάσταση στο χώρο της τεχνολογίας. Αυτό συμβαίνει διότι αυτά τα αντικείμενα μπορούν όχι μόνο τα αντιληφθούν το περιβάλλον αλλά και να επικοινωνούν με αποτέλεσμα να αποτελούν άριστα εργαλεία για την κατανόηση προβλημάτων και να αντιδρούν άμεσα σε αυτά. Ακόμα αυτά τα φυσικά συστήματα πληροφοριών εφαρμόζονται χωρίς την ανθρώπινη συμμετοχή. Ο απομακρυσμένος έλεγχος μέσω διαδικτύου μπορεί να εφαρμοστεί σε ένα μεγάλο εύρος συσκευών ή αντικειμένων προκειμένου να επιτρέψει νέες δυνατότητες ή χρήσεις του αντικειμένου, που δε θα μπορούσαν να υλοποιηθούν διαφορετικά. Ένα τέτοιο παράδειγμα, είναι ο απομακρυσμένος έλεγχος πρόσβασης σε ένα χώρο με τη δημιουργία μίας 'έξυπνης πόρτας'. Ο τυπικός έλεγχος πρόσβασης σε κοινόχρηστους χώρους, όπως εργαστήρια ή γραφεία, επιτυγχάνεται με την παράδοση κλειδιών αυτών των θυρών στους εργαζόμενους. Εντούτοις, υπάρχει το πρόβλημα της απώλειας ή κλοπής των κλειδιών, θέτοντας σε κίνδυνο την ασφάλεια του χώρου ή το πρόβλημα της πρόσβασης σε μη επιτρεπτές ώρες. Ο έξυπνος τρόπος πρόσβασης που επιτυγχάνεται μέσω διαδικτύου, επιλύει αυτά τα δύο προβλήματα και ενισχύει την ευελιξία του συστήματος ελέγχου πρόσβασης.

Η μετάβαση στην εποχή του IoT δίνει την δυνατότητα στα αντικείμενα που θα είναι δικτυωμένα να αλληλεπιδρούν μεταξύ τους. Με τον τρόπο αυτό οι χρήστες δεν χρειάζεται να αποτελούν το ενδιάμεσο μέσο που θα καλεί την υπηρεσία για να εντοπίσουν κάποιο συγκεκριμένο δεδομένο αλλά το ίδιο το αντικείμενο θα αλληλεπιδρά με το πληροφοριακό σύστημα που υπάρχει γύρω του. Έτσι πλέον η εποχή των κεντρικών υπολογιστών και των προσωπικών γίνεται ξεπερασμένη αφού η τεχνολογία θα βρίσκεται στο υπόβαθρο της καθημερινότητας με τρόπο διάφανο και αποτελεσματικό.

Οι εφαρμογές στις οποίες μπορεί να διεισδύσει το IoT είναι αμέτρητες. Για παράδειγμα ένα αυτοκίνητο θα μπορούσε να συλλέγει πληροφορίες για την κίνηση και να προτείνει εναλλακτικές διαδρομές ή να γίνεται δυναμική διαχείριση της σήμανσης για την κυκλοφοριακή αποσυμφόρηση. Για την προστασία του περιβάλλοντος θα μπορούσε να γίνεται ανίχνευση των βλαβερών σωματιδίων που υπάρχουν στην ατμόσφαιρα ή ο γρήγορος εντοπισμός πυρκαγιάς σε δασικές εκτάσεις για την έγκαιρη αντιμετώπιση. Για την καλύτερη λειτουργία στις πόλεις ο εντοπισμός διαθέσιμης θέσης parking, η έξυπνη λειτουργία για τον φωτισμό της, τα επίπεδα σκουπιδιών στους κάδους απορριμμάτων είναι κάποιες από τις πολλές εφαρμογές. Επίσης για την βελτίωση της ζωής του ανθρώπου τέτοιες εφαρμογές μπορούν να χρησιμοποιηθούν και στον τομέα της υγείας και της εκπαίδευσης. Για παράδειγμα η παρακολούθηση ασθενών σε ένα ενοποιημένο σύστημα θα μπορεί να συνδέει το ιστορικό, την φαρμακευτική αγωγή και την πορεία του ασθενή. Ακόμα πραγματοποιείται ανάπτυξη πολυμεσικών εφαρμογών και έμπειρων

συστημάτων όπως είναι η συσκευή kindle που αποτελεί ένα ηλεκτρονικό βιβλίο το οποίο προτείνει αναγνώσματα στον χρήστη ανάλογα με τις επιθυμίες του ή οποίο άλλο κριτήριο θέλει.

Η βασική και σημαντικότερη διαφορά των εφαρμογών αυτών είναι η διαφοροποίηση της διεπαφής χρήσης. Οι χρήστες πλέον θα έχουν την δυνατότητα με μεγαλύτερη ευκολία να συνδέσουν τον πραγματικό κόσμο με τον εικονικό επικοινωνώντας με φυσικότερο τρόπο τόσο με άλλους χρήστες όσο και με άλλες εφαρμογές. Ένα τέτοιος φυσικός τρόπος είναι και οι κινήσεις του σώματος. Έτσι η αναγνώριση χειρονομιών θα είναι αρκετά χρήσιμη και αποτελεσματική στον τομέα του IoT αφού ο χρήστης θα μπορεί με πολύ εύκολο τρόπο να επικοινωνήσει με τις διάφορες συσκευές που έχει ανάγκη και από τις οποίες θέλει πληροφορίες.



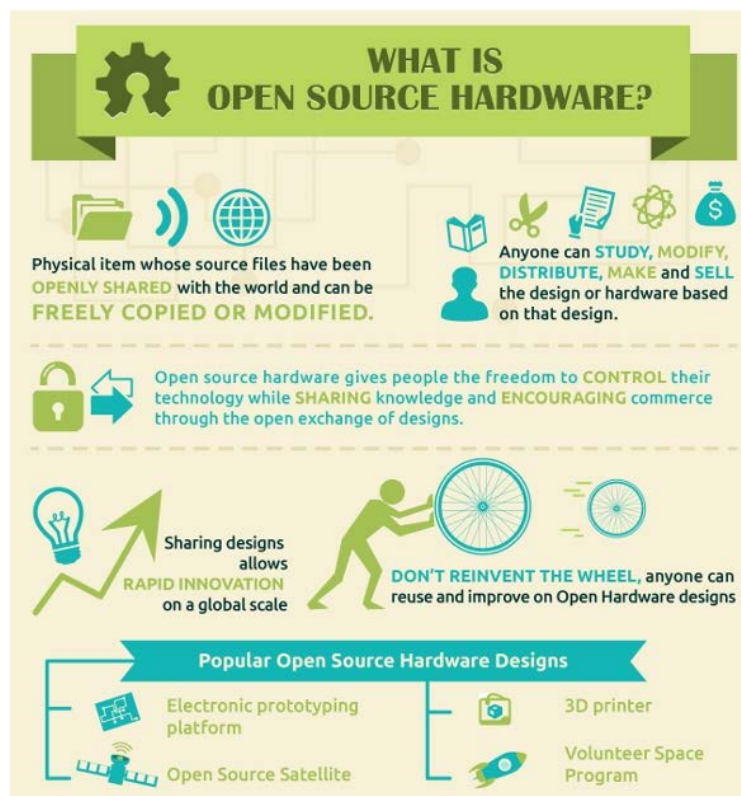
Εικόνα 6: Παραδείγματα χρήσης IoT

Το IoT γίνεται αντιληπτό ότι θα έχει μία ευρεία χρήση στην καθημερινότητα του ανθρώπου. Όμως η χρήση του πρέπει να γίνεται με προσοχή καθώς από αυτήν μπορεί να δημιουργηθούν προβλήματα ασφάλειας και έλεγχου. Είναι φανερό ότι η υλοποίηση του IoT δεν μπορεί να γίνει με τα σημερινά επίπεδα έλεγχου και ασφάλειας. Επίσης μεγάλο ζήτημα για την ανάπτυξη του αποτελεί και η ιδιωτικότητα του ανθρώπου. Έτσι πρέπει να βρεθούν τρόποι για να προστατεύονται οι πληροφορίες από τυχών αλλοιώσεις και καταστροφές καθώς και από μη εξουσιοδοτημένη χρήση των πόρων του. Επίσης πρέπει να παρέχει αξιόπιστες πληροφορίες οι οποίες να είναι διαθέσιμες στους χρήστες που τις αναζητούν. Δηλαδή πρέπει να γίνει η διασφάλιση της ακεραιότητας και της εμπιστευτικότητας των δεδομένων αλλά και της αδιάλειπτης λειτουργίας του υπολογιστικού συστήματος.

Συμπερασματικά η εποχή του IoT έρχεται να ταράξει τις ισορροπίες που ισχύουν τώρα. Οι τομείς στους οποίους θα εισχωρήσει το IoT θα βελτιώσουν την ποιότητα ζωής και θα γίνει μία πολύ καλή εξοικονόμηση ενέργειας, χρόνου και χρήματος. Το μεταβατικό στάδιο είναι αρκετά δύσκολο και για να γίνει το IoT οικείο στον άνθρωπο θα πρέπει να ξεπεραστούν όλοι οι κίνδυνοι και οι φόβοι που δημιουργεί. Όμως από την στιγμή που όλοι αυτοί οι κίνδυνοι ξεπεραστούν η ζωή του ανθρώπου θα έχει αλλάξει προς το καλύτερο.

2.4 Open Hardware

Ως Υλικό Ανοικτού Κώδικα καλούμε το σύνολο που αποτελείται τόσο από τις συσκευές που έχουν σχεδιαστεί με το πνεύμα του ανοικτού σχεδίου όσο και τα σχέδια που αφορούν αυτές (ηλεκτρομηχανολογικά, PCB, RTL σε κάποια HDL, layouts), τα οποία είναι επίσης ανοικτά. Γενικά πιο γνωστό είναι το λογισμικό ανοιχτού κώδικα όπως είναι τα Linux, Apache, Java. Εδώ και πολλά χρόνια μικρά κομμάτια από υλικό πνευματικής ιδιοκτησίας είναι διαθέσιμα ως περιγραφές σε επίπεδο RTL σε γλώσσες όπως η Verilog ή η VHDL. Σε μεγάλου μεγέθους σχέδια όμως τέτοιες περιπτώσεις ανοιχτού κώδικα δεν είναι πολύ διαδεδομένη. Ένα τέτοιο παράδειγμα μπορεί να είναι ένας μικροεπεξεργαστής. Ο πρώτος επεξεργαστής που έγινε διαθέσιμος με αυτόν τον τρόπο είναι ο LEON-32 SPARC. Στην παρακάτω εικόνα φαίνονται συνοπτικά οι κυριότερες πληροφορίες για το Open Hardware.



Εικόνα 7: Open hardware

Αντικείμενο μελέτης σε αυτόν τον τομέα αποτελεί η αναζήτηση τρόπων που θα καταστήσει το Open Hardware τόσο προσβάσιμο όσο είναι το Open Software. Οι κύριοι τρόποι σχεδιασμού του είναι ένας συνδυασμός από 3D εκτύπωση και από ανοιχτού λογισμικού ηλεκτρονικές πλατφόρμες. Μία από τις διαφορές με την δημιουργία Open Software είναι ότι το Hardware κοστίζει αρκετά χρήματα για τον σχεδιασμό και την κατασκευή του. Έτσι παρά την πρόκληση για ελαχιστοποίηση του κόστους και την μείωση των οικονομικών ρίσκων, έχουν δημιουργηθεί μοντέλα όπως το Open Source Hardware Central Bank και το KiCad, τα οποία κάνουν προσβάσιμα το Open Hardware σε περισσότερους χρήστες. Αντί να σχεδιαστούν εκ νέου άδειες ιδιοκτησίας, χρησιμοποιούνται κατά κανόνα οι ήδη υπάρχουσες από το λογισμικό.

Ένα από τα πιο γνωστά παραδείγματα του Open Hardware είναι οι πλακέτες Arduino το οποίο χρησιμοποιήθηκε και στην παρούσα διπλωματική. Το Arduino είναι μία ολοκληρωμένη ανοιχτού λογισμικού και υλικού πλατφόρμα συμπεριλαμβανομένου και του σχεδιαστικού περιβάλλοντος. Το απαραίτητο λογισμικό του Arduino είναι διαθέσιμο στο Internet χωρίς κόστος και επίσης τα σχέδια για το υλικό είναι διαθέσιμα λόγω της ανοιχτού κώδικα άδειας. Έτσι είναι φανερό ότι η κατασκευή καινούργιων πλακετών αλλά και η κατασκευή άλλων συσκευών με την βοήθεια του Arduino είναι εντελώς προσβάσιμη από τους χρήστες με ελάχιστο κόστος.

ΚΕΦΑΛΑΙΟ 3: Υλικό μέρος (Hardware)

3.1 Arduino

Το Arduino είναι μία ηλεκτρονική πλατφόρμα ανοιχτού λογισμικού, βασισμένη στον μικροελεγκτή ATmega της Atmel. Εύκολο στην χρήση και οικονομικό, το Arduino μπορεί να χρησιμοποιηθεί από τον καθένα για να φτιάξει διαδραστικά αντικείμενα και περιβάλλοντα τόσο ανεξάρτητα όσο και συνδεδεμένα με υπολογιστές μέσω προγραμμάτων. Μπορεί να προγραμματιστεί ώστε να δέχεται από το περιβάλλον δεδομένα στις εισόδους του, δηλαδή αναλογικά και ψηφιακά σήματα, να τα επεξεργάζεται και ανάλογα με την έξοδο, μπορεί να ελέγξει διάφορες συσκευές όπως leds, μηχανές, κινητήρες ή άλλους ηλεκτρονικούς επεξεργαστές. Επίσης για το Arduino έχουν κυκλοφορήσει διάφορα Arduino Shields τα οποία είναι εξαρτήματα που συνδέονται απευθείας με όλα τα pin του και επεκτείνουν σημαντικά τις λειτουργίες του. Για παράδειγμα το Arduino WiFi Shield συνδέει το Arduino στο διαδίκτυο ασύρματα.

Έχουν κατασκευαστεί αρκετά μοντέλα πλακετών Arduino τα οποία αναφέρονται στην συνέχεια ενώ για την υλοποίηση του project χρησιμοποιήθηκε η πλακέτα Arduino Uno η οποία είναι και η πιο διαδεδομένη πλακέτα Arduino.

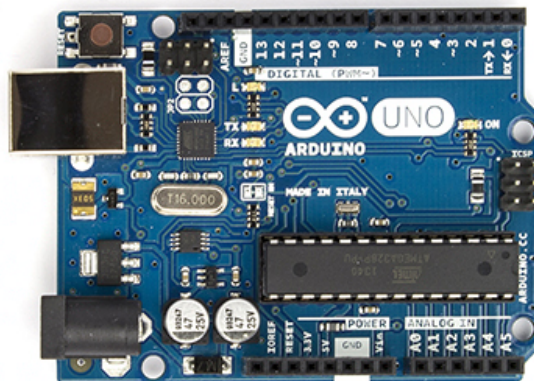
- Arduino Diecimila
- Arduino Duemilanove
- Arduino Uno
- Arduino Leonardo
- Arduino Mega
- Arduino Nano
- Arduino Due
- Arduino LilyPad
- Arduino Ethernet
- Arduino Fio
- Arduino Pro
- Arduino Pro Mini
- Arduino Mini
- Arduino Mini USB
- Arduino Bluetooth
- Arduino Serial
- Arduino Serial SS

3.2 Arduino Uno

Το Arduino Uno είναι βασισμένο στον μικροελεγκτή ATmega338. Διαθέτει 14 ψηφιακά pins και 6 αναλογικά. Επίσης 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. Η συχνότητα του είναι στα 16mhz, η οποία δεν είναι η καλύτερη που υπάρχει αλλά είναι αρκετή για τις περισσότερες εφαρμογές. Στον πίνακα φαίνονται τα βασικά χαρακτηριστικά του Arduino Uno τα οποία αναλύονται λεπτομερώς στη συνέχεια.

Microcontroller	ATmega338
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega338) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega338)
EEPROM	1 KB (ATmega338)
Clock Speed	16 MHz

Εικόνα 8: Βασικά χαρακτηριστικά Arduino Uno



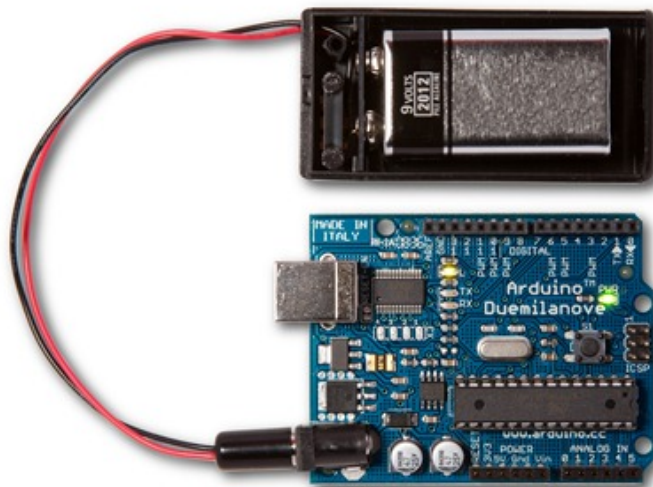
Εικόνα 9: Arduino Uno

3.2.1 Τροφοδοσία

Το Arduino Uno τροφοδοτείται μέσω σύνδεσης USB από κάποιον υπολογιστή ή μέσω εξωτερικής τροφοδοσίας από κάποιον μετασχηματιστή ή από μπαταρία. Για την σωστή λειτουργία του, η επιθυμητή τροφοδοσία πρέπει να είναι από 7 έως 12V διότι μικρότερη τροφοδοσία προκαλεί αστάθεια και μεγαλύτερη προκαλεί υπερθέρμανση και καταστροφή της πλακέτας.



Εικόνα 10: Τροφοδοσία μέσω μετασχηματιστή



Εικόνα 11: Τροφοδοσία μέσω μπαταρίας

Οι ακροδέκτες τροφοδοσίας είναι:

Vin: Σε συνδυασμό με το pin γείωσης χρησιμοποιείται ως μέθοδος τροφοδοσίας του Arduino στην περίπτωση που δεν χρησιμοποιείται εξωτερική τροφοδοσία από μετασχηματιστή ή μπαταρία. Σε αντίθετη περίπτωση το pin αυτό μπορεί να τροφοδοτήσει άλλα εξαρτήματα με τάση ίση με 7-12V δηλαδή την τάση πριν την επεξεργαστεί ο ρυθμιστής τάσης.

5V: Με το pin αυτό τροφοδοτούνται τα διάφορα εξαρτήματα με τάση 5V. Η τάση αυτή προέρχεται είτε από την θύρα USB που λειτουργεί στα 5V είτε από την εξωτερική τροφοδοσία με την βοήθεια του ρυθμιστή τάσεως.

3.3V: Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI και τροφοδοτεί με 3.3V. Δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι το όριο άντλησης ρεύματος είναι 50mA.

RESET: Όταν γειωθεί, έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.

IOREF: Παρέχει την τάση αναφοράς με την οποία λειτουργεί ο επεξεργαστής.

GND: Είσοδοι γείωσης.



Εικόνα 12: Ακροδέκτες τροφοδοσίας

3.2.2 Ακροδέκτες

Το Arduino Uno διαθέτει 14 ψηφιακούς ακροδέκτες που μπορούν να χρησιμοποιηθούν είτε σαν είσοδοι είτε σαν έξοδοι. Κάθε ακροδέκτης λειτουργεί στα 5V και μπορεί να παρέχει ή να λαμβάνει ρεύμα μεγίστης τιμής 40mA. Επίσης διαθέτει μία εσωτερική pull-up αντίσταση τιμής 20-50kOhms. Σαν έξοδος, κάποιο pin μπορεί μέσω κάποιου προγράμματος να τεθεί σε κατάσταση LOW ή HIGH και το Arduino ανάλογα με την κατάσταση να διοχετεύσει ή όχι ρεύμα σε αυτό το pin. Έτσι μπορεί να γίνει ο έλεγχος ενός led που είναι συνδεδεμένο σε αυτό. Σαν είσοδος, το Arduino μπορεί να διαβάσει την κατάσταση του pin που αλλάζει με βάση την παροχή ρεύματος από κάποια εξωτερική συσκευή, για παράδειγμα έναν διακόπτη. Κάποιοι ακροδέκτες εκτός από την I/O λειτουργία διαθέτουν και κάποιες επιπρόσθετες.

Serial 0,1: Λειτουργούν ως RX και TX αντίστοιχα, όταν ενεργοποιείται η σειριακή λειτουργία. Σε αυτήν την λειτουργία τα δεδομένα στέλνονται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB και στο pin 0, για να τα διαβάσει κάποια άλλη συσκευή. Στην περίπτωση που δέχεται δεδομένα σειριακά, αυτά είναι διαθέσιμα και στο pin 1. Με αυτό τον τρόπο το Arduino χρησιμοποιεί 2 ψηφιακές εισόδους/εξόδους λιγότερες. Το Arduino διαθέτει και 2 εξωτερικά leds RX και TX τα οποία ενεργοποιούνται αναλόγως την διαδικασία.

Serial 2,3: Λειτουργούν ως εξωτερικά interrupt. Δηλαδή μέσω ενός προγράμματος μπορούν να ρυθμιστούν ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι και να σταματάει η κανονική ροή του προγράμματος όταν συμβαίνουν συγκεκριμένες αλλαγές σε αυτές και στη συνέχεια να εκτελείται μία συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Serial 3,5,6,9,10,11: Μπορούν να λειτουργήσουν ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation). Αυτό το σύστημα διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Με αυτόν τον τρόπο, μπορεί να συνδεθεί ένα led σε κάποια από αυτά τα pin και να γίνει ο έλεγχος της φωτεινότητάς του με ανάλυση 8bit δηλαδή

256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο. Έτσι αυτά τα pin διαφοροποιούνται από τα υπόλοιπα που έχουν απλά την δυνατότητα αναμμένο-σβηστό. 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.

Serial 10,11,12,13: Οι ακροδέκτες αυτοί υποστηρίζουν SPI επικοινωνία με χρήση της βιβλιοθήκης SPI.

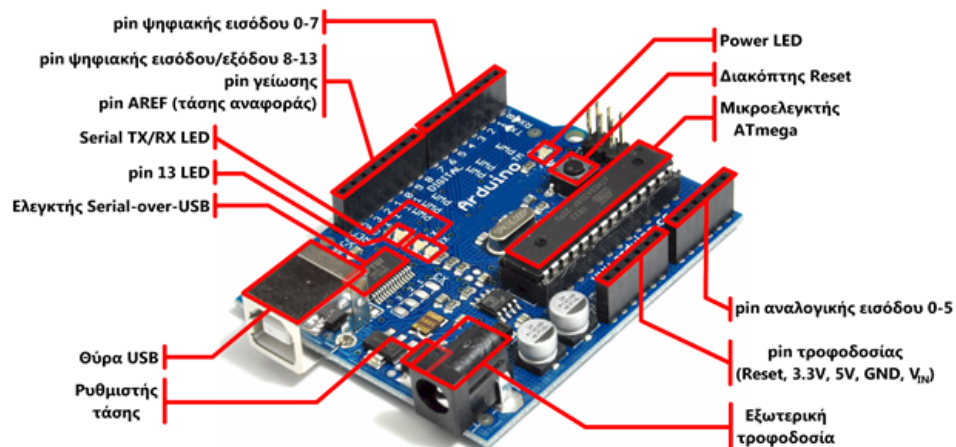
Serial 13: Σε αυτόν τον ακροδέκτη είναι συνδεδεμένο ένα ενσωματωμένο στο Arduino, Led. Όταν το pin είναι σε κατάσταση HIGH το led ανάβει και όταν είναι LOW σβήνει.



Εικόνα 13: Ψηφιακοί ακροδέκτες

Το Uno διαθέτει 6 αναλογικές εισόδους A0-A5 με 10 bit ανάλυσης δηλαδή 1024 διαφορετικές τιμές. Κανονικά μετράνε από 0 έως 5 V αλλά είναι δυνατό να μεταβληθεί το άνω άκρο μέσω του ακροδέκτη AREF και της `analogReference()`. Ο AREF είναι ένας ακροδέκτης που δίνει την τροφοδοσία αναφοράς. Η κύρια λειτουργία των αναλογικών ακροδεκτών είναι να διαβάζουν αναλογικά σήματα αλλά επίσης μπορούν να χρησιμοποιηθούν και ως ψηφιακοί. Επιπροσθέτως οι ακροδέκτες 4 και 5 μπορούν να χρησιμοποιηθούν σαν SDA και SCL αντίστοιχα σε TWI(I2C) επικοινωνία.

Όλοι οι παραπάνω ακροδέκτες φαίνονται στην παρακάτω εικόνα:



Εικόνα 14: Ακροδέκτες Arduino Uno

3.2.3 Μνήμη

Ο μικροεπεξεργαστής ATmega338 έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, στην οποία αποθηκεύονται τα Arduino sketch, SRAM (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει, και EEPROM, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών.

2KB μνήμης SRAM: Η SRAM είναι η ωφέλιμη μνήμη που χρησιμοποιείται από τα προγράμματα για την αποθήκευση μεταβλητών, πινάκων κ.α. Η μνήμη χάνει τα δεδομένα της όταν σταματήσει η παροχή ρεύματος στο Arduino ή όταν πατηθεί το κουμπί επανεκκίνησης.

1KB μνήμης EEPROM: Χρησιμοποιείται για τις εγγραφές ή τις αναγνώσεις δεδομένων από τα προγράμματα. Με την απώλεια τροφοδοσίας ή επανεκκίνησης τα περιεχόμενα δεν χάνονται.

32KB μνήμης Flash: 2 KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware είναι αναγκαίο για την εγκατάσταση προγραμμάτων στο μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.

3.3 Accelerometer

Το επιταχυνσιόμετρο είναι μία ηλεκτρονική συσκευή που έχει την ικανότητα να μετράει δυνάμεις επιτάχυνσης. Τέτοιες δυνάμεις μπορεί να είναι στατικές όπως είναι η δύναμη της βαρύτητας ή δυναμικές που είναι ευρέως γνωστές από τις κινητές συσκευές για την μέτρηση δονήσεων και κινήσεων. Με άλλα λόγια το επιταχυνσιόμετρο είναι μία συσκευή που μετράει επιταχύνσεις.

Η επιτάχυνση ορίζεται ως η μεταβολή της ταχύτητας ως προς τον χρόνο. Για παράδειγμα ένα αμάξι επιταχυνόμενο από την ακινησία μέχρι μία ταχύτητα 60mph σε 6 sec έχει μία επιτάχυνση τιμής 10mph ανά sec. Επίσης το επιταχυνσιόμετρο μετράει και την επιτάχυνση που συνδέεται με τις δυνάμεις βάρους. Για παράδειγμα ένα επιταχυνσιόμετρο σε ηρεμία στην επιφάνεια της γης θα μετρήσει επιτάχυνση $g = 9.81\text{m/s}^2$ λόγω του βάρους του. Αντιθέτως ένα επιταχυνσιόμετρο το οποίο βρίσκεται είτε σε ελεύθερη πτώση ή σε ηρεμία στο διάστημα θα μετρήσει τιμή ίση με το μηδέν.

3.3.1 Εφαρμογές

Μετρώντας την στατική επιτάχυνση που προκαλείται από την βαρύτητα , το επιταχυνσιόμετρο μπορεί να μετρήσει την κλίση της γωνίας κάποιας συσκευής σε σχέση με την γη. Καταγράφοντας τη δυναμική επιτάχυνση μπορεί να αναλυθεί το πώς κινείται η συσκευή και να αντιληφθεί το περιβάλλον της καλύτερα. Με τον κατάλληλο κώδικα μπορούν να απαντηθούν πολλά ερωτήματα όπως αν κινείται ή βρίσκεται οριζόντια ή κάθετα , τι θα συμβεί στο επόμενο βήμα και πολλά άλλα.

Στην καθημερινότητα τα επιταχυνσιόμετρα είναι ευρέως διαδεδομένα. Στα κινητά τηλέφωνα και σε άλλες συσκευές χρησιμοποιούνται σε πολλές εφαρμογές ελέγχοντας την θέση της συσκευής. Στα laptop τα επιταχυνσιόμετρα χρησιμοποιούνται ώστε να εντοπίζουν κάποια απότομη πτώση και ο σκληρός δίσκος να σβήνει ώστε αποφευχθεί η απώλεια δεδομένων. Με παρόμοιο τρόπο εφαρμόζεται στα αυτοκίνητα η χρήση των αερόσακων σε περίπτωση ατυχημάτων.

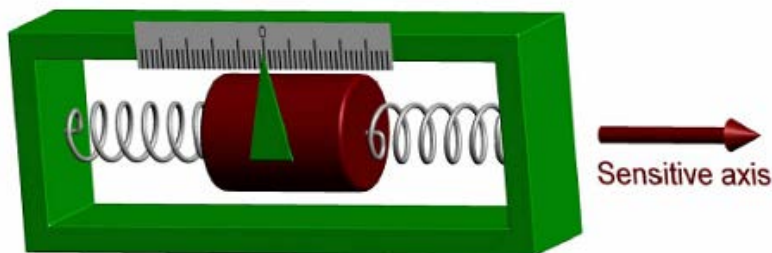
3.3.2 Λειτουργία

Η αρχή λειτουργίας του επιταχυνσιόμετρου βασίζεται στην τοποθέτηση μίας μάζας στο εσωτερικό τους συνδεδεμένη με ένα σύστημα ελατηρίων. Οι δυνάμεις που αναπτύσσονται μετατοπίζουν την μάζα είτε στην αρνητική είτε στην θετική κατεύθυνση και παράλληλα συσπειρώνουν ή επιμηκώνουν το ελατήριο. Η μεταβολή των ελατηρίων είναι σύμφωνα με τον νόμο του Hook ανάλογη με την αναπτυσσόμενη δύναμη. Και από τον δεύτερο νόμο του Νεύτωνα προσδιορίζεται η επιτάχυνση της μάζας.

$$F_k = -k \cdot x$$

Η επιτάχυνση αυτή όμως δεν είναι μόνο αυτή που οφείλεται στην κίνηση του σώματος αλλά περιέχει και άλλες επιδράσεις με κυριότερη την επιτάχυνση της βαρύτητας. Έτσι αν είναι γνωστή η επιτάχυνση της βαρύτητας σε μία περιοχή τότε μπορεί να προσδιοριστεί η πραγματική επιτάχυνση του σώματος.

$$a = F / m + g = f + g$$



Εικόνα 15: Λειτουργία επιταχυνσιόμετρου

Τα επιταχυνσιόμετρα χωρίζονται σε δύο ομάδες. Όλα χρησιμοποιούν την μάζα και την παραπάνω λειτουργία, εκείνο που διαφέρει είναι ο τρόπος που μετράται η μετατόπιση της μάζας και μετατρέπεται σε ένδειξη επιτάχυνσης. Στα ανοιχτού κύκλου γίνεται ο προσδιορισμός της δύναμης που μετατοπίζει την μάζα από την θέση ισορροπίας ενώ στα κλειστού τύπου της δύναμης που απαιτείται για να διατηρηθεί η μάζα στην θέση ισορροπίας.

3.4 Γυροσκόπιο

Τα γυροσκόπια είναι συσκευές οι οποίες χρησιμοποιούν τις αρχές της αδρανείας και της διατήρησης της στροφορμής για να διατηρήσουν σταθερό προσανατολισμό τους σε σχέση με μία αρχική καθορισμένη διεύθυνση. Έτσι είναι ιδανικές συσκευές για την μέτρηση των γωνιακών περιστροφών και κατ'επέκταση τον προσδιορισμό των διορθώσεων που πρέπει να επιβληθούν στους φορείς τους ώστε οι τελευταίοι να διατηρήσουν σταθερή διεύθυνση. Τα γυροσκόπια μπορούν να ταξινομηθούν με βάση την αρχή λειτουργίας και τα κατασκευαστικά χαρακτηριστικά τους σε τρεις μεγάλες κατηγορίες τα μηχανικά, οπτικά και τα μικρο ηλεκτρομηχανικά.

Τα γυροσκόπια είναι οι σιωπηλοί «εγκέφαλοι» που κρατούν τα αεροπλάνα στον αέρα, τους δορυφόρους στην τροχιά τους και τα πλοία στην πορεία τους. Τα πρώτα πρακτικά γυροσκόπια εφευρέθηκαν το 1910 από τον E.Sperry, η εταιρία του οποίου κατασκεύασε και τους πρώτους σταθεροποιητές αεροπλάνων. Από τότε και μέχρι σήμερα γύρω στις 400 πατέντες για γυροσκόπια έχουν μετατραπεί σε συστήματα αυτόματης πλοήγησης και οδήγησης για πολιτικά και στρατιωτικά αεροσκάφη, πυραύλους, βόμβες, δορυφόρους και διαστημόπλοια. Σήμερα τα γυροσκόπια οδηγούν ρομπότ, αντλιοσθητικά συστήματα αυτοκινήτων, το Διαστημικό Τηλεσκόπιο Hubble, το Διεθνές Διαστημικό Σταθμό, διαστημικά οχήματα που κινούνται στην επιφάνεια του Άρη, ακόμα και τα συστήματα αυτοπροώθησης των αστροναυτών όταν κάνουν περίπατο στο Διάστημα.

Τα γυροσκόπια λειτουργούν με βάση δύο αρχές. Σύμφωνα με την αρχή της διατήρησης της στροφορμής, μία περιστρεφόμενη μάζα διατηρεί σταθερή την κατεύθυνση του άξονα περιστροφής. Ένα περιστρεφόμενο γυροσκόπιο κάνει ένα δορυφόρο να είναι σταθερός και πάντοτε στραμμένος προς τη Γη. Μία περιστρεφόμενη μάζα επίσης αντιστέκεται στις δυνάμεις που προσπαθούν να αλλάξουν την κατεύθυνση του άξονα περιστροφής. Στα μέσα του περασμένου αιώνα, τεράστια γυροσκόπια, που ζύγιζαν τόνους, βιδώνονταν στην άτρακτο των πλοίων και περιστρέφονταν από κινητήρες ώστε να πιέζουν αντίθετα όταν τα κύματα έτειναν να ανατρέψουν τα σκάφη.

Σύμφωνα με την αρχή της μετάπτωσης, ένα περιστρεφόμενο γυροσκόπιο με ελευθερία μετακίνησης κινείται κάθετα ως προς τις εξωτερικές δυνάμεις που του ασκούνται. Μία παιδική σβούρα δεν ανατρέπεται όταν τη σπρώξουμε ελαφρά, αλλά μετακινείται στο πλάι. Ένα

γυροσκόπιο σε αεροπλάνο που πετάει παράλληλα με το έδαφος θα παρουσιάσει μετάπτωση όταν τα φτερά αρχίσουν να παίρνουν κλίση. Ανιχνεύοντάς την τα όργανα δείχνουν στον πιλότο τη γωνία που έχουν τα φτερά ως προς τον ορίζοντα, στοιχείο καθοριστικής σημασίας ιδίως όταν η πτήση γίνεται στα τυφλά. Ένα δεύτερο γυροσκόπιο μπορεί να δείξει την κλίση της ατράκτου του αεροπλάνου προς τα μπρος ή προς τα πίσω. Συνδεδεμένα με επιταχυνσιόμετρα τα γυροσκόπια μπορούν να πετάξουν σταθερά το αεροπλάνο, λειτουργία γνωστή ως αυτόματος πιλότος.

Όλο και λιγότερα γυροσκόπια είναι μηχανικά όσο περνάει ο καιρός. Στη δεκαετία του 1980, τα γυροσκόπια δαχτυλιδιού λέιζερ και τα γυροσκόπια με οπτικές ίνες αποδείχτηκαν πιο ακριβή και ταυτόχρονα πιο μικρά και ελαφριά. Η τιμή τους είναι γύρω στα 4.000\$. Ορισμένες εταιρίες κατασκευάζουν μικροσκοπικά γυροσκόπια πάνω σε φύλλα πυριτίου, που αν και λιγότερο ακριβή μπορούν να παραχθούν μαζικά, όπως τα ολοκληρωμένα κυκλώματα και στοιχίζουν μόλις 20\$, επιτρέποντας τη χρήση τους ακόμα και σε μεσαίας ή χαμηλής αξίας αυτοκίνητα.

3.5 MPU 6050

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε το MPU 6050 το οποίο είναι μία πλακέτα 6 βαθμών ελευθερίας που περιέχει ένα επιταχυνσιόμετρο τριών διαστάσεων και ένα γυροσκόπιο επίσης τριών διαστάσεων. Η τροφοδοσία του είναι μία τάση τιμής 3.3V και η επικοινωνία του γίνεται μέσω I2C με μέγιστη ταχύτητα 400kHz.

Τα βασικά χαρακτηριστικά του mpu είναι:

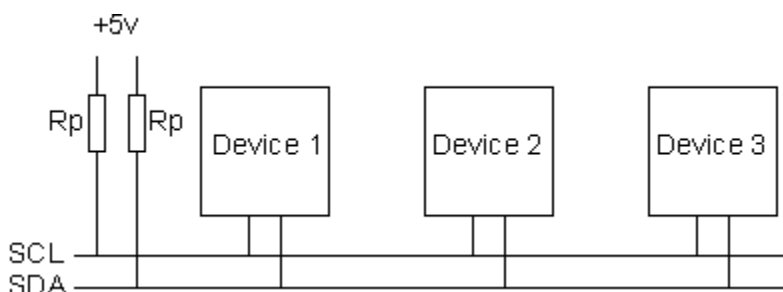
- Επιλογή εύρους του επιταχυνσιόμετρου σε $\pm 2/4/8/16g$
- Επιλογή εύρους του γυροσκόπιου σε $\pm 250/500/1000/2000$ degrees/s
- 1 I2C bus που κάνει το mpu “slave” σε μία άλλη συσκευή όπως ένα Arduino
- 1 I2C bus που κάνει το mpu “master” σε μία άλλη συσκευή όπως ένα μαγνητόμετρο
- 16 bit έξοδος και για τους δύο αισθητήρες
- Ευαισθησία γυροσκόπιου στα 0.1 degrees/s
- Χαμηλός θόρυβος και στις δύο εξόδους
- Ρυθμός δεδομένων εξόδων στα 1000Hz



Εικόνα 16: MPU 6050

3.5.1 I2C bus

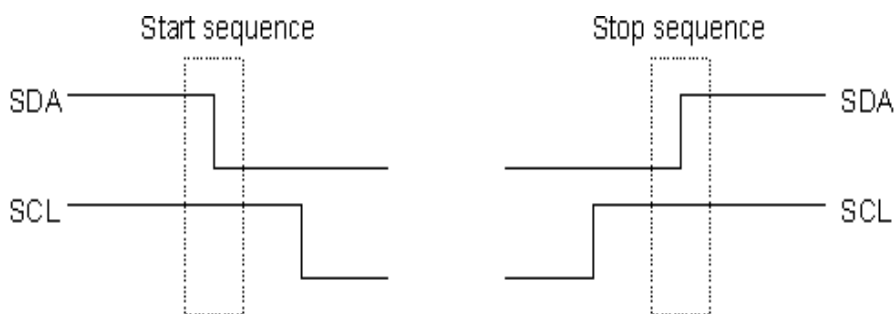
Ο I2C αποτελείται από δύο καλώδια τα SCL και SDA. Το SCL χρησιμοποιείται για να συγχρονίσει την μεταφορά δεδομένων. Το SDA είναι η γραμμή δεδομένων. Πρέπει να υπάρχει και μία τρίτη γραμμή η οποία είναι η γείωση ενώ μπορεί να υπάρχει και άλλη μία για 5V τροφοδοσίας των συσκευών. Τα SCL και SDA μπορούν να έχουν σαν έξοδο LOW αλλά όχι HIGH. Για να έχουμε στην έξοδο HIGH πρέπει να υπάρχουν pull-up αντιστάσεις στην γραμμή των 5V. Πρέπει να υπάρχουν 2 pull-up αντιστάσεις, η μία να συνδέει το SDA και η άλλη το SCL με τα 5V.



Εικόνα 17: Στοιχεία I2C bus

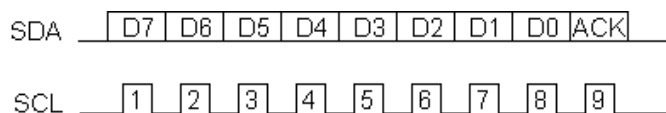
Οι συσκευές που είναι συνδεδεμένες μέσω του I2C είναι είτε masters ή slaves. Ο master είναι πάντα η συσκευή που οδηγεί την γραμμή του SCL. Τα slaves είναι οι συσκευές που απαντούν στον master. Ένα slave δεν μπορεί να αρχίσει την μεταφορά δεδομένων μέσω του bus αλλά μόνο ο master. Συνήθως υπάρχει μόνο ένας master στην γραμμή και πολλοί slaves. Η πολλαπλή χρήση masters είναι δυνατή αλλά ασυνήθιστη. Και τα δύο μπορούν να μεταφέρουν δεδομένα αλλά η μεταφορά πάντα ελέγχεται και ξεκινά από τον master.

Όταν ο master θέλει να ξεκινήσει μία μεταφορά δεδομένων τότε υλοποιεί μία ακολουθία έναρξης. Η άλλη ακολουθία που μπορεί να υλοποιήσει είναι η ακολουθία τερματισμού. Όταν γίνεται μεταφορά δεδομένων η γραμμή δεδομένων πρέπει να είναι σταθερή και να μην αλλάζει ενώ το SCL να είναι HIGH. Όμως αυτές οι δύο ακολουθίες αποτελούν εξαίρεση διότι το SDA μπορεί να αλλάζει. Η ακολουθία έναρξης και τερματισμού σηματοδοτούν την αρχή και το τέλος μία επικοινωνίας με την slave συσκευή.



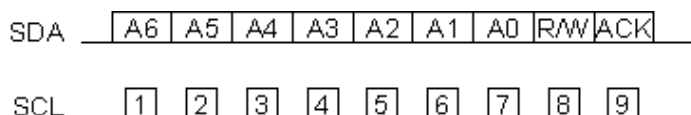
Εικόνα 18: Ακολουθία έναρξης και τερματισμού

Τα δεδομένα μεταφέρονται σε ακολουθίες των 8 bits. Τα bits αυτά τοποθετούνται στην γραμμή SDA ξεκινώντας με το MSB (Most Significant Bit). Η γραμμή SCL γίνεται HIGH και μετά LOW. Για κάθε ένα από τα 8 bits που μεταφέρονται η συσκευή που τα λαμβάνει στέλνει πίσω ένα bit. Αν είναι LOW τότε σημαίνει ότι έλαβε το bit και είναι έτοιμο να αποδεχτεί το επόμενο bit. Αν στείλει HIGH τότε δεν μπορεί να δεχτεί άλλα δεδομένα και έτσι τερματίζεται η λειτουργία στέλνοντας μία ακολουθία τερματισμού. Η συνηθισμένη ταχύτητα του ρολογιού του bus φτάνει τα 100KHz. Υπάρχουν μοντέλα που φτάνουν ταχύτητες από 400KHz έως 3.4MHz



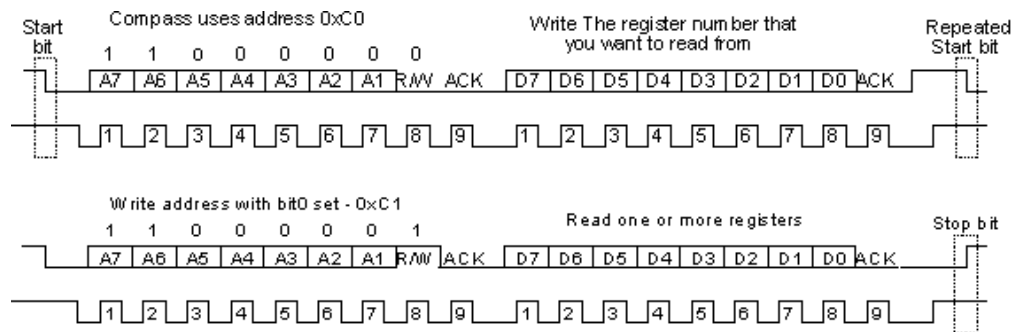
Εικονα 19: Δεδομένα στον I2C bus

Οι διευθύνσεις του I2C είναι είτε 7 bits ή 10 bits. Η χρήση των 10 bits είναι σπάνια ενώ των 7 bits πιο διαδεδομένη. Έτσι είναι φανερό ότι σε ένα I2C bus μπορούν να υπάρχουν μέχρι 128 συσκευές. Όταν στέλνεται αυτή η διεύθυνση παρόλο που είναι 7 bit ,στέλνονται 8bits. Αυτό το επιπλέον bit χρησιμοποιείται για να πληροφορήσει τον slave αν ο master διαβάζει ή γράφει σε αυτόν. Αν το bit είναι 0 ο master γράφει στον slave ενώ αν είναι 1 διαβάζει.



Εικόνα 20: Διευθύνσεις I2C bus

Η πρώτη ενέργεια που πραγματοποιείται είναι ότι ο master στέλνει την ακολουθία έναρξης. Αυτή η ενέργεια θα ειδοποιήσει τους slaves στο bus ότι ξεκινά μία συναλλαγή ώστε να είναι έτοιμοι να λάβουν δεδομένα σε περίπτωση που είναι για αυτούς. Στη συνέχεια στέλνεται η διεύθυνση της συσκευής. Ο slave με αυτή την διεύθυνση θα συνεχίσει να περιμένει τα δεδομένα ενώ οι υπόλοιποι θα αγνοήσουν την διαδικασία μέχρι την επόμενη αποστολή. Ο master μετά στέλνει τον αριθμό του καταχωρητή του slave στον οποίο θέλει να γράψει. Αφού γίνουν όλα τα παραπάνω τότε γίνεται και η αποστολή των δεδομένων. Η αποστολή θα συνεχιστεί μέχρι να σταλούν τα δεδομένα και στη συνέχεια ο master στέλνει την ακολουθία τερματισμού ώστε να σταματήσει η διαδικασία. Σε περίπτωση read η διαδικασία ξεκινά σαν την διαδικασία write επειδή πρέπει να είναι γνωστή η διεύθυνση της συσκευής. Αφού σταλεί αυτή η διεύθυνση , ο master στέλνει ξανά μία ακολουθία έναρξης και την διεύθυνση της συσκευής αλλά αυτή τη φορά με το τελευταίο bit HIGH.



Εικόνα 21: Διαδικασίες write και read

Όταν ο master διαβάζει από τον slave, ο slave είναι αυτός που τοποθετεί τα δεδομένα στην SDA γραμμή. Όμως όταν ο slave δεν είναι έτοιμος να στείλει τα δεδομένα έχουμε επιπλοκές. Με συσκευές όπως είναι οι EEPROMs αυτό δεν δημιουργεί επιπλοκές όμως αν η slave συσκευή είναι ένας μικροεπεξεργαστής με πολλά πράγματα να κάνει τότε δημιουργείται πρόβλημα. Ο μικροεπεξεργαστής θα πρέπει να καλέσει μία interrupt routine, να αποθηκεύσει τους τρέχοντες καταχωρητές, να βρει την διεύθυνση του master και να πάρει τα δεδομένα. Αυτή η διαδικασία μπορεί να διαρκέσει πολύ χρόνο και παράλληλα ο master στέλνει σήματα στην SCL γραμμή που ο slave δεν μπορεί να απαντήσει. Το I2C παρέχει μία λύση σε αυτή την επιπλοκή δίνοντας την δυνατότητα στον slave να μπορεί να κρατήσει την SCL γραμμή LOW. Όταν δέχεται εντολή διαβάσματος τότε διατηρεί την γραμμή LOW και ο μικροεπεξεργαστής παίρνει τα δεδομένα που χρειάζονται και τα τοποθετεί στους καταχωρητές εκπομπής. Στη συνέχεια αφήνει την γραμμή επιτρέποντας στην pull-up αντίσταση να την μεταβάλει σε HIGH. Από την πλευρά του master, θα στείλει την εντολή για το read κάνοντας την SCL HIGH και περιμένει να δει αν θα παραμείνει HIGH. Αν είναι LOW τότε ο slave δεν είναι έτοιμος και ο master περιμένει

3.6 Bluetooth

Το Bluetooth είναι μία ασύρματη τεχνολογία η οποία κατασκευάστηκε με σκοπό να αντικαταστήσει τα καλώδια κατά τις συνδέσεις ηλεκτρονικών συσκευών όπως τα κινητά τηλέφωνα και τα laptop. Το Bluetooth δουλεύει στα 2.4 GHz που είναι το μη αδειοδοτημένο φάσμα ISM το οποίο είναι ανοιχτό στον οποιοδήποτε. Έτσι σε αυτό το φάσμα είναι φανερό ότι λόγω της πληθώρας συσκευών δημιουργείται πρόβλημα.

Το Bluetooth για να αντιμετωπίσει τέτοιου είδους προβλήματα χρησιμοποιεί την τεχνολογία FHSS (Frequency Spectrum Hopping Technology) που σημαίνει ότι δύο συσκευές που είναι συνδεδεμένες με Bluetooth και ανταλλάσσουν δεδομένα μετά την αποστολή κάθε πακέτου αλλάζουν περιοδικά κανάλι επικοινωνίας. Αυτό σημαίνει ότι αν ένα κανάλι είναι δεσμευμένο από κάποια άλλη συσκευή, η συσκευή Bluetooth θα αλλάξει αυτόματα κανάλι για να μην επηρεαστεί η επικοινωνία. Η τεχνολογία αυτή επίσης διασφαλίζει την ύπαρξη πολλών δικτύων Bluetooth ταυτόχρονα χωρίς να επηρεάζονται οι συνδέσεις τους.

Οι συσκευές που είναι συνδεδεμένες με Bluetooth μπορούν να δημιουργήσουν δίκτυα γνωστά ως piconets, που αποτελούνται από δύο έως οκτώ συσκευές. Από αυτές τις συσκευές η μία είναι υπεύθυνη για την διεύθυνση της κυκλοφορίας δεδομένων σε όλο το δίκτυο και ονομάζεται master. Οι υπόλοιπες συσκευές λειτουργούν ως δευτερεύουσες και ονομάζονται slaves. Η κύρια συσκευή αποτελεί το κέντρο του δικτύου και παίρνει όλες τις αποφάσεις με αποτέλεσμα να ορίζει και το συνολικό εύρος ζώνης του δικτύου. Όταν μία ή περισσότερες συσκευές του piconet συνδεθούν σε ένα άλλο piconet, τότε έχουμε ένα scatternet. Οι συσκευές Bluetooth μπορεί να συμπεριφέρονται σαν slaves σε διάφορα piconet, αλλά master μπορούν να είναι αυστηρά μόνο σε ένα. Οι συσκευές αυτές που ανήκουν σε πολλά piconet μπορούν να δράσουν σαν συνδετικός κρίκος μεταξύ των υποδικτύων για τη μεταφορά δεδομένων.



Εικόνα 22: Bluetooth

Οι συσκευές Bluetooth συνήθως έχουν εμβέλεια περίπου 10 μέτρα. Υπάρχουν και ισχυρές συσκευές με εμβέλεια 100 μέτρα. Γενικά είναι μία τεχνολογία με μικρή εμβέλεια σε σχέση με άλλες τεχνολογίες κατηγορίας IEEE 802.11. Παρόλα αυτά η μικρή εμβέλεια αποτελεί πλεονέκτημα κάποιες φορές. Για παράδειγμα στα κινητά τηλεφώνά η μικρή εμβέλεια αποτελεί ασφάλεια για τον χρήστη.

3.6.1 Bluetooth mate

Το Bluetooth Mate είναι ένα εργαλείο το οποίο μπορεί να αντικαταστήσει τα σειριακά καλώδια πολύ αποτελεσματικά. Αυτές οι συσκευές χρησιμοποιούν την σειριακή λειτουργία RX TX. Η μεταφορά δεδομένων είναι πολύ εύκολη και γρήγορη 2400 μέχρι 115200bps και λειτουργούν μέχρι τα 350ft. Το Bluetooth mate περιέχει ρυθμιστές τάσης ώστε να μπορεί να λειτουργήσει από τροφοδοσία 3.3V έως 6V.

Κάποια από τα βασικά χαρακτηριστικά του είναι:

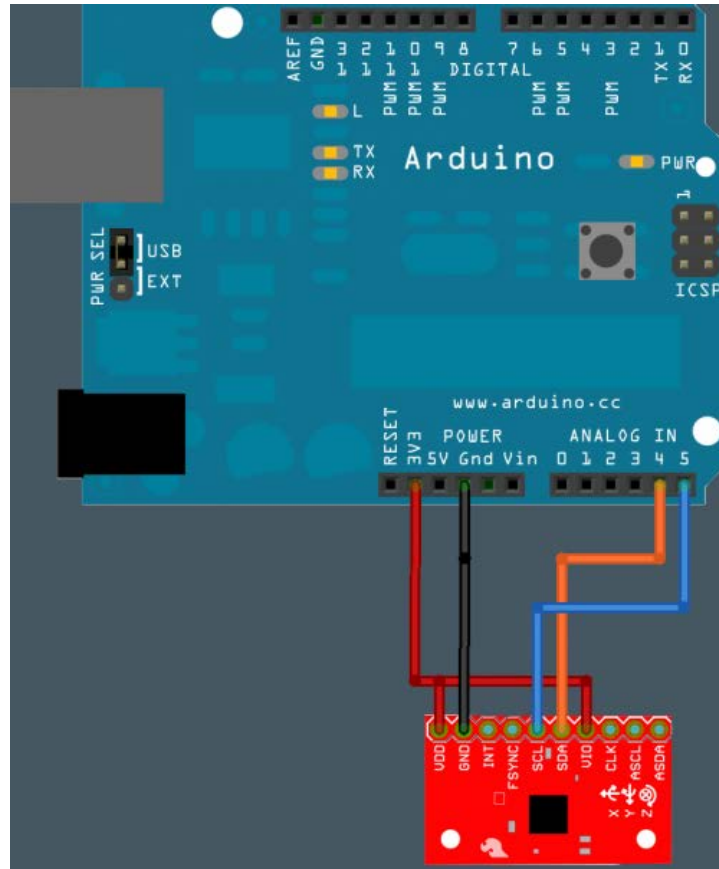
- v6.15 Firmware
- Μεγάλη απόσταση μετάδοσης
- Μικρή κατανάλωση ρεύματος 25mA
- Λειτουργία σε πολυσύχναστα περιβάλλοντα όπως WiFi, 802.11g χωρίς παρεμβολές
- Συχνότητα 2.4~2.524 GHz
- Τάση λειτουργίας 3.3V-6V
- Σειριακή επικοινωνία 2400-115200bps
- Θερμοκρασία λειτουργίας -40 ~ +70C
- Ενσωματωμένη κεραία
- Σχεδιασμένο με τέτοιο τρόπο ώστε να συνδέεται απευθείας με κάποια μοντέλα Arduino



Εικόνα 23: Bluetooth mate

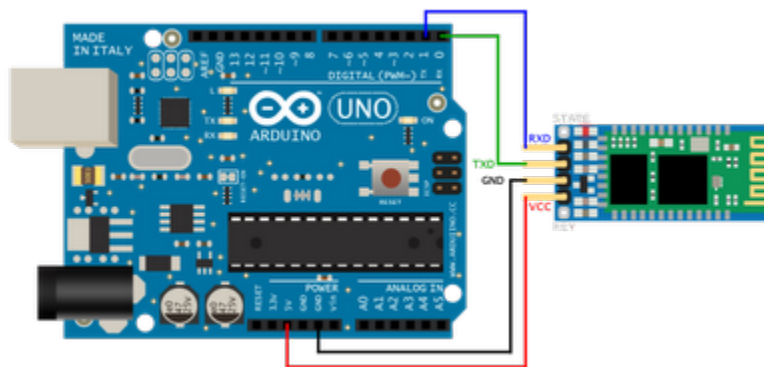
3.7 Περιγραφή συνδέσεων

Αρχικά γίνεται η σύνδεση του Arduino με το mpu 6050. Η σύνδεση είναι απλή. Τα μόνα pins από το mpu που πρέπει να συνδεθούν είναι το GRD στο GRD του Arduino, το VDD και το VIO στα 3.3V και οι ακροδέκτες SCL και SDA στους ακροδέκτες A5 και A4 του Arduino αντίστοιχα. Η σύνδεση αυτή γίνεται γιατί όπως έχει αναφερθεί στην περιγραφή του Arduino, οι ακροδέκτες 4 και 5 μπορούν να χρησιμοποιηθούν σαν SDA και SCL αντίστοιχα σε TWI(I2C) επικοινωνία. Οι ακροδέκτες ASCL και ASDA δεν χρησιμοποιούνται διότι αποτελούν έναν δεύτερο bus I2C ο οποίος δεν χρειάζεται στην παρούσα εργασία. Οι συνδέσεις φαίνονται αναλυτικά στο παρακάτω σχήμα.



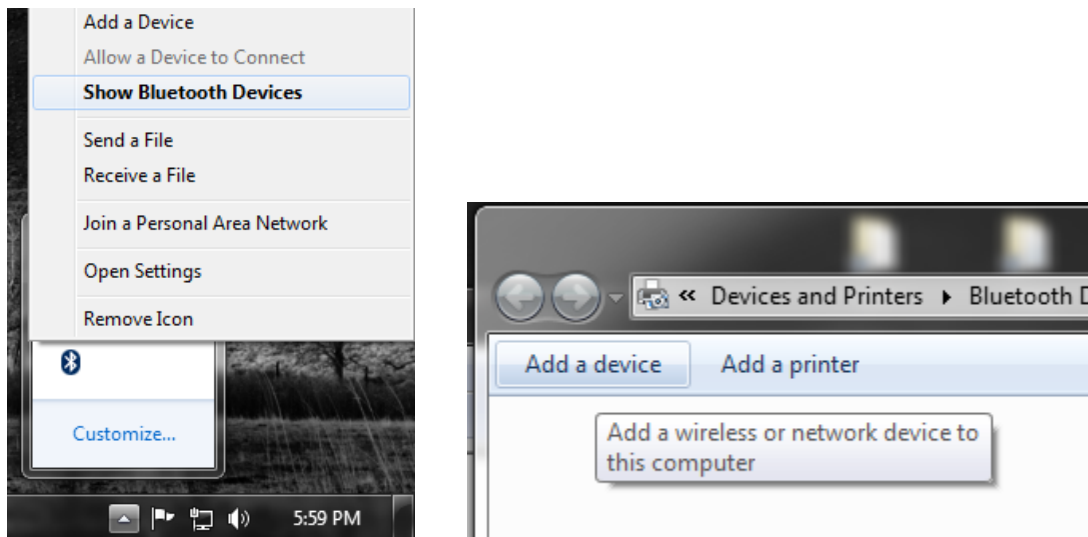
Εικόνα 24: Σύνδεση Arduino-Mpu

Στη συνέχεια πρέπει να γίνει η σύνδεση με το Bluetooth για την ασύρματη λειτουργία της συσκευής. Και πάλι η σύνδεση είναι απλή. Τα pins που πρέπει να συνδεθούν είναι τα έξης. Το GRD του Bluetooth με το GRD του Arduino, το Vcc με την τάση 3.3V και τα TX,RX με τα RX,TX τα οποία είναι τα digital pin 0,1 του Arduino αντίστοιχα. Αφού γίνουν οι συνδέσεις και ξεκινήσει η τροφοδοσία του Arduino, στο Bluetooth mate θα αρχίσει να αναβοσβήνει ένα κόκκινο λαμπάκι που σημαίνει ότι είναι ενεργό. Η τροφοδοσία έγινε με την βοήθεια μπαταρίας 9V.



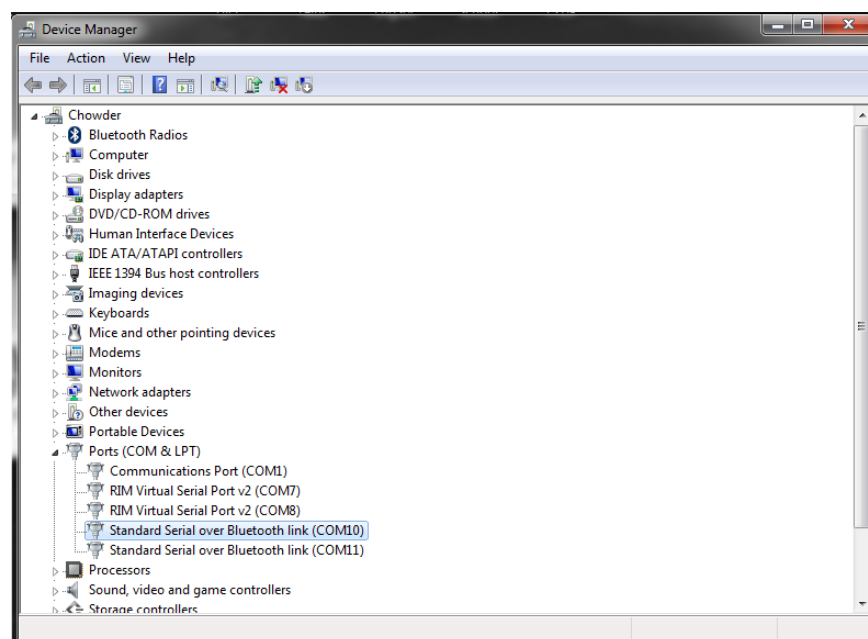
Εικόνα 25: Σύνδεση Arduino-Bluetooth

Αφού ολοκληρωθούν οι συνδέσεις πρέπει να γίνει η αναγνώριση της συσκευής Bluetooth από τον υπολογιστή. Σε περίπτωση που ο υπολογιστής δεν διαθέτει Bluetooth υπάρχουν συσκευές που προσθέτουν αυτή την λειτουργία. Αφού ενεργοποιηθεί το Bluetooth τότε πρέπει από τον πίνακα ελέγχου να γίνει η επιλογή add a device ώστε να γίνει η εγκατάσταση του. Κατά την διάρκεια της εγκατάστασης θα ζητηθεί passkey το οποίο για λόγους ευκολίας είναι 1234.



Εικόνα 26: Εγκατάσταση Bluetooth

Μετά την εγκατάσταση στον device manager στον τομέα Ports φαίνεται ποια πύλη καταλαμβάνει η συσκευή η οποία θα χρησιμοποιηθεί αργότερα. (COM). Αφού ολοκληρωθούν όλα τα παραπάνω βήματα η συσκευή είναι έτοιμη από άποψη υλικού.









Εικόνα 27: Εύρεση COM

ΚΕΦΑΛΑΙΟ 4: Λογισμικό μέρος (Software)

4.1 Arduino IDE

Η διαχείριση του Arduino από τον υπολογιστή γίνεται μέσω του Arduino IDE το οποίο είναι διαθέσιμο στην επίσημη ιστοσελίδα του Arduino για τα βασικότερα λειτουργικά συστήματα όπως τα Windows, τα Mac και τα Linux. Το λογισμικό αυτό είναι ένα πολύ πρακτικό περιβάλλον για την δημιουργία των προγραμμάτων του Arduino τα οποία ονομάζονται sketches και περιέχουν χρωματική σήμανση. Το λογισμικό περιέχει αρκετά έτοιμα παραδείγματα για την ευκολότερη κατανόηση της λειτουργίας του, βιβλιοθήκες για την επέκταση της γλώσσας και για την λειτουργία εξαρτημάτων συνδεδεμένα με το Arduino, τον compiler για την μεταγλώττιση των προγραμμάτων, ένα serial monitor για την παρακολούθηση της σειριακής επικοινωνίας και τέλος την επιλογή για φόρτωση του προγράμματος στο Arduino. Στον παρακάτω πίνακα φαίνονται τα σημαντικότερα κουμπιά του IDE και οι λειτουργίες τους.

	Verify	Ελέγχει για συντακτικά λάθη στον κώδικα.
	Upload	Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino.
	New	Δημιουργεί ένα νέο sketch.
	Open	Παραθέτει ένα μενού με όλα τα sketch. Κάνοντας κλικ σε ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.
	Save	Αποθηκεύει ένα sketch.
	Serial Monitor	Ανοίγει την σειριακή οθόνη.

Εικόνα 28: Βασικές λειτουργίες Arduino IDE

Η γλώσσα που χρησιμοποιεί το Arduino IDE είναι βασισμένη στην γλώσσα Wiring, μία παραλλαγή C/C++ για μικροελεγκτές AVR όπως ο ATmega. Υποστηρίζει όλες τις βασικές δομές της C καθώς και ορισμένα χαρακτηριστικά της C++. Ο Compiler που χρησιμοποιείται είναι ο AVR gcc και η βασική βιβλιοθήκη C είναι η AVR libc. Οι εντολές και συναρτήσεις που χρησιμοποιούνται είναι ίδιες με τις βασικές της C με την ίδια σύνταξη, ίδιους τύπους δεδομένων και ίδιους τελεστές. Επιπλέον υπάρχουν και κάποιες ειδικές εντολές, συναρτήσεις και σταθερές για την διαχείριση του ειδικού hardware του Arduino. Κάποιες από αυτές φαίνονται στον παρακάτω πίνακα.

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	-	(<i>pin, mode</i>)	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	(pin)	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	-	(pin, value)	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	-	(time)	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 μίας delay.
detachInterrupt	Εντολή	-	(interrupt)	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .

noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των interrupt
interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μία εντολή noInterrupts.
Serial.begin	Μέθοδος κλάσης	-	(<i>datarate</i>)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	-	(<i>data</i>)	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

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

// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

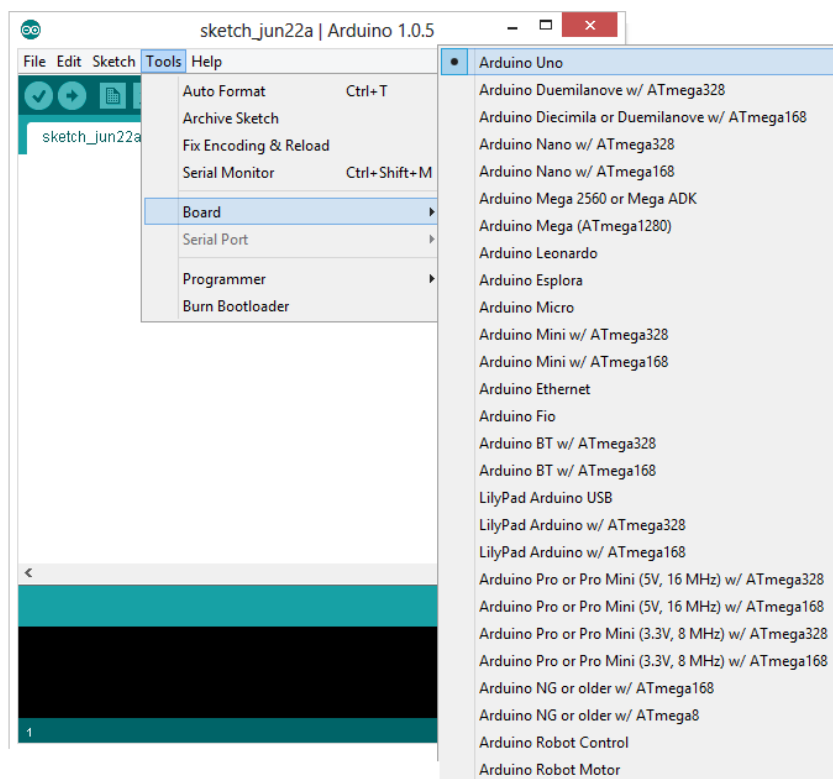
```
void setup()
{
  // ...
}
```

```
void loop()
{
  // ...
}
```

// Υπόλοιπες συναρτήσεις...

Η βασική ρουτίνα setup() εκτελείται μία 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).

Το Arduino IDE είναι εύκολο στην εγκατάσταση και χρήση του. Το μόνο που πρέπει να γίνει πριν τη συγγραφή του κώδικα είναι η επιλογή της πλακέτας καθώς και η επιλογή της Serial Port που χρησιμοποιείται.



Εικόνα 30: Επιλογή σωστής πλακέτας

4.2 Θόρυβος και φίλτρα

Ένα από τα βασικά προβλήματα στην συλλογή δεδομένων είναι ο θόρυβος. Μία από τις πλέον επιθυμητές ιδιότητες ενός οργάνου μετρήσεων μίας φυσικής ή χημικής ποσότητας, είναι η ικανότητα διάκρισης της πραγματικής τιμής της ποσότητας μέσα από ανεπιθύμητα σήματα. Σήμα είναι ο φορέας στον οποίο κωδικοποιείται η ζητούμενη πληροφορία κατά τη διαδικασία μίας μέτρησης ενώ ο θόρυβος ορίζεται ως το σύνολο των ανεπιθύμητων πληροφοριών, οι οποίες συνοδεύουν το σήμα και υποβαθμίζουν την ορθότητα και την ακρίβεια μίας μέτρησης.

Σε πολλές περιπτώσεις ο θόρυβος έχει καθορισμένη προέλευση και ιδιαίτερα χαρακτηριστικά, η γνώση των οποίων μπορεί να συμβάλει αποτελεσματικά στην αντιμετώπιση του. Ένας τρόπος για την μείωση του θορύβου αποτελούν τα φίλτρα τα οποία έχουν την δυνατότητα να επεξεργάζονται τα δεδομένα που μας ενδιαφέρουν ώστε να είναι πιο ακριβή χωρίς ανεπιθύμητες συνιστώσες.

Στη συγκεκριμένη εφαρμογή οι μετρήσεις του επιταχυνσιόμετρου περιλαμβάνουν περισσότερα από το διάνυσμα της βαρύτητας. Οποιαδήποτε μικρή δύναμη που ασκείται στο αντικείμενο μεταβάλλει την μέτρηση του επιταχυνσιόμετρου. Έτσι είναι ασταθές βραχυπρόθεσμα. Από την άλλη πλευρά το γυροσκόπιο μπορεί να κάνει ακριβείς μετρήσεις βραχυπρόθεσμα όμως όταν το σύστημα επιστρέφει στην αρχική του θέση το γυροσκόπιο δεν επιστρέφει μηδενικές τιμές. Έτσι δεν είναι αξιόπιστο μακροπρόθεσμα. Για αυτούς τους λόγους ο συνδυασμός των οργάνων με ένα φίλτρο μπορεί να δώσει αξιόπιστες μετρήσεις.

4.2.1 Kalman Filter

Το Kalman Filter είναι ένας αλγόριθμος ο οποίος χρησιμοποιεί μία σειρά από μετρήσεις που έχουν γίνει σε ένα χρονικό διάστημα. Αυτές οι μετρήσεις περιέχουν θόρυβο ο οποίος συνεισφέρει σε λάθη. Ο αλγόριθμος προσπαθεί να υπολογίσει την κατάσταση του συστήματος με βάση την παρούσα και τις προηγούμενες καταστάσεις και έτσι η μέτρηση θα είναι πιο ακριβής. Δηλαδή είναι ένας βέλτιστος, προς κάθε τεκμηριωμένο κριτήριο λειτουργικότητας, αναδρομικός αλγόριθμος επεξεργασίας δεδομένων. Κατά αυτήν την έννοια, ο αλγόριθμος λαμβάνει υπόψη οποιαδήποτε πληροφορία σχετική με το σύστημα μπορεί να του παρασχεθεί και επεξεργάζεται αυτήν την πληροφορία μαζί με κάθε μέτρηση, ανεξαρτήτως της ακρίβειάς της, για να εκτιμήσει όσο το δυνατόν καλύτερα τις τρέχουσες τιμές των μεταβλητών που ενδιαφέρουν το σύστημα.

Ο όρος αναδρομικός αλγόριθμος ή φίλτρο, αναφέρεται στην ιδιότητα του να χρησιμοποιεί προηγούμενες πληροφορίες χωρίς να χρειάζεται όλη η προσφερόμενη πληροφορία να αποθηκευτεί για να επεξεργαστεί ξανά κάθε φορά που μια καινούρια μέτρηση παρουσιάζεται. Συχνά όμως σε ένα σύστημα, οι μεταβλητές οι οποίες περιγράφουν την κατάστασή του, συμβαίνει να μην μπορούν να μετρηθούν είτε άμεσα είτε αξιόπιστα. Κατά συνέπεια, είναι απαραίτητο να μπορούν να εξαχθούν συμπερασματικά οι τιμές αυτών των μεταβλητών από οποιαδήποτε σχετική διαθέσιμη πληροφορία για αυτές. Εκτός των άλλων, είναι γνωστό πως σε κάθε μέτρηση εισάγεται ως ένα βαθμό κάποιο σφάλμα. Αυτό το σφάλμα προκύπτει συνήθως ως συνάρτηση, θορύβου προερχόμενου από το ίδιο το σύστημα ελέγχου και φυσικά σφαλμάτων που εισάγουν οι ίδιες οι συσκευές μέτρησης. Όλα τα παραπάνω λοιπόν, τονίζουν την αναγκαιότητα ύπαρξης κάποιου φίλτρου, ικανού να εξαλείφει κάθε παράγοντα αποπροσανατολισμού ή παραμόρφωσης της εκτίμησης της κατάστασης του συστήματος.

Ο αλγόριθμος λειτουργεί σε δύο βήματα. Στο πρώτο, το οποίο είναι το στάδιο της πρόβλεψης, το φίλτρο παράγει εκτιμήσεις για της τωρινές καταστάσεις καθώς και για τα σφάλματά τους. Μόλις η τιμή της επόμενης κατάστασης παρατηρηθεί μαζί με το σφάλμα που αναγκαστικά υπάρχει τότε αυτές οι εκτιμήσεις ανανεώνονται με περισσότερο βάρος να δίνεται στις εκτιμήσεις που έχουν μεγαλύτερη βεβαιότητα. Αυτό είναι το δεύτερο στάδιο.

Το Kalman Filter λειτουργεί παράγοντας στατιστικές εκτιμήσεις βασισμένες στις μετρήσεις που γίνονται. Για να γίνει αυτό πρέπει να είναι γνωστός τόσο και ο θόρυβος που εισάγεται στο φίλτρο όσο και ο θόρυβος όλου του συστήματος. Αυτό μπορεί να γίνει αν ο θόρυβος μπορεί να μοντελοποιηθεί σε γκαουσιανή-κανονική κατανομή πράγμα που είναι εφικτό αφού ο τυχαίος θόρυβος έχει αυτό το χαρακτηριστικό. Η κανονική κατανομή αναφέρεται σε συνεχείς μεταβλητές αποτελώντας μία συνεχή συνάρτηση πυκνότητας πιθανότητας. Χρησιμοποιείται ως μία πρώτη προσέγγιση για να περιγραφούν τυχαίες μεταβλητές πραγματικών τιμών, οι οποίες τείνουν να συγκεντρώνονται γύρω από μία μέση τιμή. Η συνάρτηση που περιγράφει αυτή την κατανομή είναι η εξής:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

4.2.2 Μαθηματική ανάλυση Kalman Filter

Για την καλύτερη κατανόηση του φίλτρου αρχικά παρουσιάζονται οι βασικές έννοιες και οι βασικές εξισώσεις που είναι απαραίτητες για την υλοποίηση του.

Η προηγούμενη εκτιμώμενη κατάσταση βασίζεται στην προηγούμενη κατάσταση και τις εκτιμήσεις των καταστάσεων πριν από αυτήν.

$$\hat{\mathbf{x}}_{k-1|k-1}$$

A priori είναι η εκτίμηση του πίνακα καταστάσεων στην παρούσα κατάσταση βασισμένη στις προηγούμενες καταστάσεις του συστήματος και στις εκτιμήσεις πριν από αυτήν:

$$\hat{\mathbf{x}}_{k|k-1}$$

A posteriori είναι η εκτίμηση της κατάστασης μίας δεδομένης στιγμής k με βάση τις παρατηρήσεις μέχρι εκείνη την στιγμή συμπεριλαμβανομένης και της στιγμής k:

$$\hat{\mathbf{x}}_{k|k}$$

Το πρόβλημα που δημιουργείται είναι ότι οι καταστάσεις του συστήματος πρέπει να μελετηθούν μέσω παρατηρήσεων z_k . Το σύστημα αυτό ονομάζεται Hidden Markov Model. Αυτό σημαίνει ότι μία κατάσταση θα βασίζεται στην κατάσταση στον χρόνο k και σε όλες τις προηγούμενες

καταστάσεις. Επιπλέον αυτό έχει σαν αποτέλεσμα η εκτίμηση της κατάστασης να μην είναι έμπιστη πριν το Kalman σταθεροποιηθεί.

Το \hat{x} αναφέρεται στην εκτίμηση μίας κατάστασης ενώ το x στην κανονική κατάσταση. Η κατάσταση του συστήματος μία δεδομένη στιγμή k δίνεται από την σχέση :

$$x_k = Fx_{k-1} + Bu_k + w_k$$

Όπου x_k είναι ο πίνακας καταστάσεων ο οποίος είναι :

$$x_k = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k$$

Η έξοδος του φίλτρου είναι η γωνία θ και επίσης η θ_b η οποία είναι η ποσότητα κατά την οποία το γυροσκόπιο δημιουργεί σφάλμα. Έτσι η πραγματική τιμή δίνεται αν αφαιρεθεί από την μέτρηση του γυροσκόπιου, το σφάλμα.

Ο πίνακας F είναι το μοντέλο μετάβασης το οποίο εφαρμόζεται στην κατάσταση x_{k-1} .

$$F = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}$$

Το u_k είναι η είσοδος ελέγχου που στην παρούσα περίπτωση είναι η μέτρηση του γυροσκοπίου σε μοίρες ανά δευτερόλεπτο. Αυτή η μέτρηση γράφεται ως θ_k . Έτσι η εξίσωση μπορεί να γραφτεί με αυτόν τον τρόπο:

$$x_k = Fx_{k-1} + B\dot{\theta}_k + w_k$$

Ο πίνακας B είναι το μοντέλο της εισόδου ελέγχου. Ορίζεται ως:

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}$$

Οι τιμές είναι απόλυτα φυσιολογικές διότι πολλαπλασιάζοντας το Δt με το $\dot{\theta}$ παίρνουμε την γωνιακή θέση θ και επειδή το σφάλμα θ_b δεν υπολογίζεται απευθείας από το $\dot{\theta}$ το δεύτερο στοιχείο του πίνακα είναι μηδέν.

Ο θόρυβος w_k είναι γκαουσιανής κατανομής και έχει την μορφή:

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$$

Ο πίνακας \mathbf{Q}_k περιέχει τις τιμές σφαλμάτων του επιταχυνσιόμετρου και του γυροσκοπίου και είναι της μορφής:

$$\mathbf{Q}_k = \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t$$

Όπως φαίνεται ο πίνακας εξαρτάται από τον χρόνο k και η τιμή του επιταχυνσιόμετρου και το σφάλμα του γυροσκοπίου πολλαπλασιάζονται με το Δt . Αυτό έχει νόημα διότι όσο περνά ο χρόνος ο θόρυβος αυξάνεται.

Αυτές οι τιμές πρέπει να είναι γνωστές για να δουλέψει το φίλτρο και αναλόγως την τιμή τους το φίλτρο μπορεί να δουλεύει σωστά ή όχι. Για παράδειγμα αν οι εκτιμήσεις είναι αργές τότε εμπιστευόμαστε παραπάνω από όσο πρέπει το επιταχυνσιόμετρο και πρέπει να μειωθεί η τιμή του Q_θ για καλύτερα αποτελέσματα.

Η μέτρηση του z_k της κατάστασης x_k είναι:

$$z_k = \mathbf{H} x_k + v_k$$

Η τιμή της είναι ίση με την τωρινή κατάσταση πολλαπλασιασμένη με έναν πίνακα \mathbf{H} συν την μέτρηση του θορύβου v_k . Ο πίνακας \mathbf{H} είναι το μοντέλο παρατήρησης. Αφού η μέτρηση προέρχεται μόνο από το επιταχυνσιόμετρο ο πίνακας είναι:

$$\mathbf{H} = [1 \quad 0]$$

Ο θόρυβος της μέτρησης είναι γκαουσιανής κατανομής και γράφεται:

$$\mathbf{v}_k \sim N(0, \mathbf{R})$$

Ο \mathbf{R} δεν είναι πίνακας και είναι μόνο μία μεταβλητή και ορίζεται:

$$\mathbf{R} = E \begin{bmatrix} v_k & v_k^T \end{bmatrix} = \text{var}(v_k)$$

Υποθέτουμε ότι ο θόρυβος είναι ίδιος και δεν μεταβάλλεται με τον χρόνο, άρα:

$$\text{var}(v_k) = \text{var}(v)$$

Αν η μεταβλητή είναι πολύ υψηλή το φίλτρο θα λειτουργεί με πολύ αργό ρυθμό αφού θα εμπιστεύεται νέες τιμές λιγότερο. Αντιθέτως αν είναι πολύ μικρή, οι μετρήσεις μπορεί να περιέχουν πολύ θόρυβο αφού θα εμπιστευόμαστε το επιταχυνσιόμετρο σε μεγάλο βαθμό.

Έτσι χρειαζόμαστε τις τιμές των Q_θ , $Q_{\dot{\theta}_b}$ και $\text{var}(v)$.

Παρακάτω φαίνονται οι εξισώσεις για τον υπολογισμό της κατάστασης του συστήματος \hat{x}_k σε δύο φάσεις, την πρόβλεψη και το update.

Πρόβλεψη

Με τις δύο πρώτες εξισώσεις γίνεται προσπάθεια πρόβλεψης της τωρινής κατάστασης και του σφάλματος την στιγμή k . Αρχικά το φίλτρο υπολογίζει την τωρινή κατάσταση βασισμένο στις προηγούμενες καταστάσεις και στην μέτρηση του γυροσκοπίου.

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + B\dot{\theta}_k$$

Για αυτό το λόγο ονομάζεται είσοδος ελέγχου διότι χρησιμοποιείτε ως μία επιπλέον είσοδος για να εκτιμήσει την παρούσα κατάσταση που ονομάζεται a priori state $\hat{x}_{k|k-1}$ όπως έχει αναφερθεί. Στη συνέχεια υπολογίζεται το a priori σφάλμα $P_{k|k-1}$ βασισμένο στο προηγούμενο σφάλμα on the $P_{k-1|k-1}$, το οποίο ορίζεται ως:

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k$$

Αυτός ο πίνακας χρησιμοποιείται για να γίνει αντιληπτό πόσο πολύ αξιόπιστες είναι οι τωρινές τιμές της εκτιμώμενης κατάστασης. Όσο μικρότερος είναι τόσο περισσότερο αξιόπιστη είναι η τωρινή εκτιμώμενη κατάσταση. Είναι φανερό ότι το σφάλμα αυξάνεται από το προηγούμενο update αφού το σφάλμα πολλαπλασιάζεται με τους πίνακες F , F^T και προσθέτουμε τον θόρυβο Q_k της στιγμής k .

Ο πίνακας P στην συγκεκριμένη περίπτωση είναι ένας 2×2 πίνακας:

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}$$

Update

Για το update αρχικά γίνεται ο υπολογισμός της διαφοράς ανάμεσα στην μέτρηση z_k και στην a priori κατάσταση $\hat{x}_{k|k-1}$.

$$\tilde{y}_k = z_k - H\hat{x}_{k|k-1}$$

Στη συνέχεια γίνεται έλεγχος για το κατά πόσο αξιόπιστες είναι οι μετρήσεις με βάση το a priori σφάλμα και τον πίνακα R . Όσο μεγαλύτερη είναι η τιμή της μέτρησης του θορύβου τόσο μεγαλύτερη είναι και η τιμή του S . Αυτό σημαίνει πως οι εισερχόμενες μετρήσεις δεν είναι αξιόπιστες.

Στη συνέχεια υπολογίζουμε την παρακάτω ποσότητα:

$$S_k = HP_{k|k-1}H^T + R$$

Αυτή προσπαθεί να προβλέψει κατά πόσο μπορούμε να εμπιστευτούμε την μέτρηση η οποία είναι βασισμένο στον a priori πίνακα σφάλματος $P_{k|k-1}$ και στον πίνακα R . Όσο μεγαλύτερη είναι η μέτρηση του θορύβου τόσο μεγάλη είναι και η τιμή S και σημαίνει ότι δεν μπορούμε να εμπιστευτούμε την εισερχόμενη μέτρηση.

Στη συνέχεια γίνεται ο υπολογισμός του κέρδους του φίλτρου.

$$K_k = P_{k|k-1}H^T S_k^{-1}$$

Αν δεν γνωρίζουμε τις αρχικές τιμές του συστήματος τότε ο πίνακας P είναι:

$$P = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}$$

Οπου L είναι ένας μεγάλος αριθμός.

Στην συγκεκριμένη εφαρμογή επειδή υποθέτουμε ότι το αρχικό σφάλμα του γυροσκοπίου είναι 0 και επειδή βρίσκουμε την αρχική γωνιακή θέση έχει τεθεί:

$$\mathbf{P} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Σε αυτήν την περίπτωση το κέρδος του φίλτρου είναι ένας 2×1 πίνακας:

$$\mathbf{K} = \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}$$

Μετά από τα παραπάνω μπορεί να γίνει το update της a posteriori εκτίμησης της τωρινής κατάστασης:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

Αυτό γίνεται προσθέτοντας της a priori κατάστασης $\hat{\mathbf{x}}_{k|k-1}$ με το κέρδος του φίλτρου πολλαπλασιασμένο κατά $\tilde{\mathbf{y}}_k$ το οποίο είναι η διαφορά της μέτρησης \mathbf{z}_k και της εκτιμώμενης a priori κατάστασης $\hat{\mathbf{x}}_{k|k-1}$. Άρα μπορεί να είναι είτε θετικό ή αρνητικό.

Τέλος ανανεώνεται ο a posteriori πίνακας σφάλματος:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1}$$

Όπου \mathbf{I} είναι ο πίνακας ταυτότητας και ορίζεται ως:

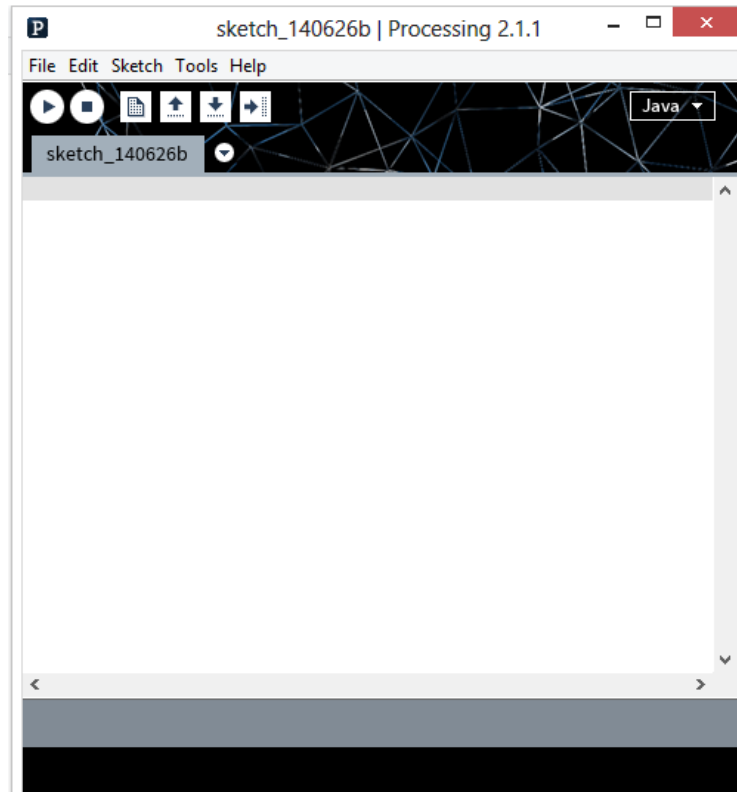
$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Αυτό που κάνει το φίλτρο δηλαδή είναι ότι διορθώνει τον πίνακα σφαλμάτων με βάση τις διορθώσεις των εκτιμήσεων

4.3 Processing

Το Processing είναι μία γλώσσα προγραμματισμού ανοικτού κώδικα και παράλληλα ένα προγραμματιστικό περιβάλλον για ανθρώπους που θέλουν να προγραμματίσουν εικόνες, animation και ήχο. Σκοπός του είναι η εκμάθηση προγραμματισμού από αρχάριους χρήστες μέσω ενός οπτικού πλαισίου καθώς και η παροχή ενός επαγγελματικού εργαλείου παραγωγής πολυμεσικών εφαρμογών. Σήμερα υπάρχουν χιλιάδες μαθητές, σχεδιαστές και ερευνητές που χρησιμοποιούν το processing για εκπαίδευση και δημιουργία προγραμμάτων ή εφαρμογών.

Το περιβάλλον του Processing είναι γραμμένο σε Java. Προγράμματα που έχουν γραφτεί σε Processing είναι επίσης μεταφρασμένα σε Java και έπειτα τρέχουν σαν Java προγράμματα. Προγράμματα που είναι γραμμένα σε Java και Processing, συνήθως τρέχουν γρηγορότερα από προγράμματα που βασίζονται σε scripting γλώσσες όπως ActionScript και Lingo, κάτι το οποίο είναι πολύ σημαντικό για εφαρμογές με γραφικά. Μεγάλες διακρίσεις μεταξύ του Processing και της Java είναι οι βιβλιοθήκες γραφικών του Processing και το απλοποιημένο στυλ προγραμματισμού που δεν προϋποθέτει οι χρήστες να καταλαβαίνουν πιο προηγμένες έννοιες όπως κλάσεις, αντικείμενα, ή animation (ενώ εξακολουθούν να είναι διαθέσιμες για προχωρημένους χρήστες). Τέτοιες τεχνικές λεπτομέρειες πρέπει να είναι ειδικά προγραμματισμένες σε Java, αλλά ολοκληρωμένες σε Processing, κάνοντας τα προγράμματα μικρότερα και ευκολότερα να διαβαστούν.



Εικόνα 31: Processing

Μερικά βασικά χαρακτηριστικά του Processing είναι:

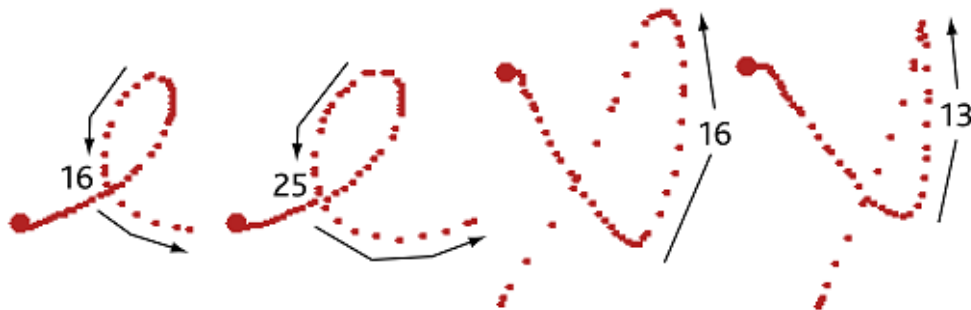
- Ανοιχτού λογισμικού και δωρεάν εφαρμογή
- Διαδραστικά προγράμματα με 2D, 3D και PDF έξοδο
- Συμβατό με GNU/Linux, Mac OS και Windows
- Πάνω από 100 βιβλιοθήκες που επεκτείνουν τις λειτουργίες του
- Μεγάλη βιβλιογραφία και πολλοί οδηγοί για την κατανόηση της λειτουργίας του

Το Processing αποτελεί ένα εναλλακτικό εργαλείο από τα εργαλεία με ακριβές άδειες για αυτό και είναι προσιτό σε σχολεία και μαθητές. Επειδή είναι ανοιχτού κώδικα συνέχεια προστίθενται από τους χρήστες του βιβλιοθήκες και άλλα εργαλεία που είναι χρήσιμα όπως είναι η παρουσίαση δεδομένων με γραφικό τρόπο, η σύνθεση μουσικής, η σύνδεση με το διαδίκτυο και ο προγραμματισμός ηλεκτρονικών συσκευών.

Η χρήση του Processing όπως είπαμε μπορεί να δημιουργήσει εφαρμογές που παρουσιάζουν οπτικά ένα αποτέλεσμα. Για αυτό το λόγο έγινε και η χρήση του σε αυτή την εργασία ώστε να γίνεται αντιληπτό το αποτέλεσμα της κίνησης και της χειρονομίας μέσω ενός περιβάλλοντος που είναι πιο προσιτό και πιο κατανοητό στον χρήστη. Σε αυτό βοήθησε και το γεγονός ότι το Arduino χρησιμοποιεί ένα περιβάλλον που είναι παρόμοιο στο περιβάλλον του Processing.

4.4 1\$ Unistroke αλγόριθμος αναγνώρισης

Μία κίνηση για την πραγματοποίηση μία χειρονομίας είναι ένα σύνολο από υποψήφια σημεία C και πρέπει να βρεθεί ποιο από τα προηγούμενα αποθηκευμένα σχήματα T μοιάζει περισσότερο σε αυτό. Αυτά τα σημεία συνήθως αποκτούνται με την χρήση μίας διακριτικής συσκευής η οποία δημιουργεί ένα μονοπάτι. Όμως λόγω σφαλμάτων αλλά και από τον διαφορετικό τρόπο που ο χρήστης μπορεί να μετακινήσει την συσκευή αυτά τα δύο σχήματα σπανίως ταιριάζουν απολύτως μεταξύ τους. Για παράδειγμα στο παρακάτω σχήμα ο αριθμός των σημείων, η ταχύτητα και κλίση διαφέρουν αλλά όλα αναφέρονται στο ίδιο σχήμα. Αυτό το χαρακτηριστικό είναι μία πρόκληση για την αναγνώριση των χειρονομιών.



Εικόνα 32: Διαφορετικός αριθμών σημείων για το ίδιο σχήμα

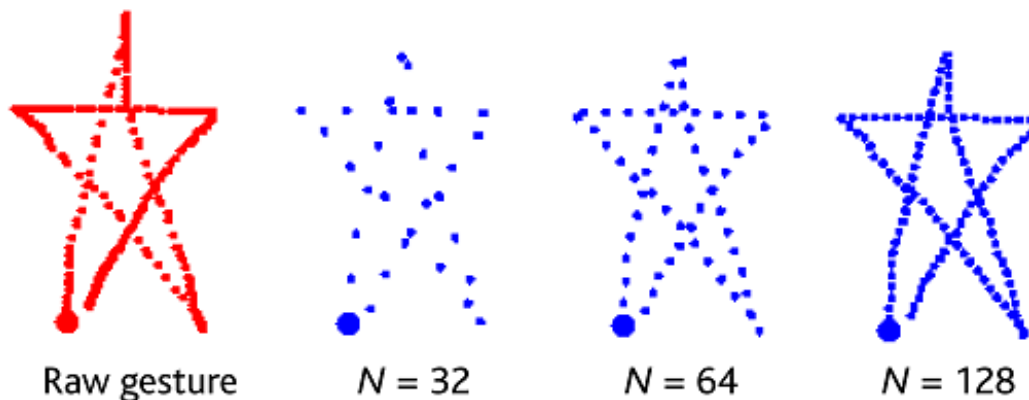
Ο αλγόριθμος πρέπει να διαθέτει ορισμένα χαρακτηριστικά:

- Πρέπει να μπορεί να εντοπίζει το είδος του σχήματος παρόλο την διαφορετικότητα της ταχύτητας και της αίσθησης.
- Να μπορεί να εντοπίσει το σχήμα ανεξαρτήτως μεγέθους και κλίσης που έχει.
- Να μην περιλαμβάνει περίπλοκες μαθηματικές έννοιες και τεχνικές όπως αντιστροφές πινάκων.
- Ο κώδικας να είναι εύκολος και κατανοητός σε λίγες γραμμές.
- Αποδοτικός και γρήγορος.
- Να είναι εύκολη η εισαγωγή νέων σχημάτων για αναγνώριση.
- Να δίνει το σωστό σχήμα ανεξαρτήτως του αριθμού των σημείων εισαγωγής.

Ο αλγόριθμος περιλαμβάνει τέσσερα βασικά βήματα από τα οποία περνάνε όλα τα σχήματα προς αναγνώριση. Αυτά είναι τα βήματα της επαναδειγματοληψίας, περιστροφής, αλλαγής κλίμακας και μετάφρασης. Τα υποψήφια σημεία C μετά την επεξεργασία συγκρίνονται με τα σημεία T_i των αποθηκευμένων σχημάτων για να βρεθεί το σωστό. Τα βήματα που αναφέρθηκαν περιγράφονται παρακάτω.

Βήμα 1

Όπως έχει αναφερθεί η δειγματοληψία των χειρονομιών στις διεπαφές χρήστη γίνεται με ρυθμό που καθορίζεται από το hardware και το software. Έτσι η ταχύτητα της κίνησης επηρεάζει τον αριθμό των σημείων εισόδου της χειρονομίας. Για να γίνουν οι διάφορες διαδρομές της κίνησης συγκρίσιμες, ακόμα και σε διαφορετικές ταχύτητες, πρέπει να γίνει επαναδειγματοληψία της χειρονομίας έτσι ώστε η αρχική διαδρομή που ορίζεται από M σημεία να ορίζεται από N ισαπέχοντα σημεία. Χρησιμοποιώντας ένα πολύ χαμηλό N υπάρχει περίπτωση απώλειας ακρίβειας της διαδικασίας ενώ αντιθέτως μεγάλο N προσθέτει πολύ χρόνο στις συγκρίσεις των διαδρομών. Μετά από πειραματικές μετρήσεις οι τιμές $32 \leq N \leq 256$ είναι επαρκείς και χρησιμοποιήθηκε η τιμή $N=64$.



Εικόνα 33: Επαναδειγματοληψία

Η διαδικασία αυτή γίνεται υπολογίζοντας πρώτα την συνολική απόσταση του μονοπατιού που αποτελείται από τα M σημεία. Διαιρώντας με $N-1$ βρίσκεται η απόσταση I που θα απέχουν τα N σημεία μεταξύ τους. Έτσι δημιουργώντας το νέο μονοπάτι, όταν ξεπεραστεί η τιμή I προσθέτεται ένα νέο σημείο. Στο τέλος της διαδικασίας τα σημεία θα είναι σύνολο N . Αυτό επιτρέπει να υπολογιστεί η απόσταση $C[k]$ έως $Ti[k]$ για $k=1$ μέχρι N .

RESAMPLE(*points*, *n*)

```

1   $I \leftarrow \text{PATH-LENGTH}(\text{points}) / (n - 1)$ 
2   $D \leftarrow 0$ 
3   $\text{newPoints} \leftarrow \text{points}_0$ 
4  foreach point  $p_i$  for  $i \geq 1$  in points do
5       $d \leftarrow \text{DISTANCE}(p_{i-1}, p_i)$ 
6      if  $(D + d) \geq I$  then
7           $q_x \leftarrow p_{i-1_x} + ((I - D) / d) \times (p_{i_x} - p_{i-1_x})$ 
8           $q_y \leftarrow p_{i-1_y} + ((I - D) / d) \times (p_{i_y} - p_{i-1_y})$ 
9          APPEND(newPoints,  $q$ )
10         INSERT(points,  $i$ ,  $q$ ) //  $q$  will be the next  $p_i$ 
11          $D \leftarrow 0$ 
12     else  $D \leftarrow D + d$ 
13 return newPoints
```

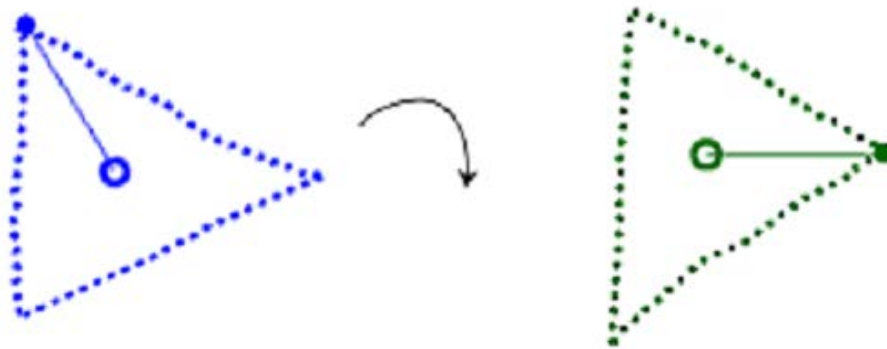
PATH-LENGTH(*A*)

```

1   $d \leftarrow 0$ 
2  for  $i$  from 1 to  $|A|$  step 1 do
3       $d \leftarrow d + \text{DISTANCE}(A_{i-1}, A_i)$ 
4  return  $d$ 
```

Βήμα 2

Ένα πρόβλημα που υπάρχει είναι ότι δύο ίδιες χειρονομίες που αναγνωρίζονται από το ίδιο σχήμα μπορεί να έχουν πραγματοποιηθεί χρησιμοποιώντας διαφορετική γωνία. Έτσι πρέπει να βρεθεί η βέλτιστη γωνία ώστε να περιστραφεί το σχήμα για να γίνει η αναγνώριση του. Αυτό γίνεται υπολογίζοντας την γωνία που σχηματίζεται μεταξύ του κέντρου της χειρονομίας και της γωνίας του πρώτου σημείου της. Στη συνέχεια περιστρέφεται ώστε αυτή η γωνία να είναι στις 0 μοίρες.



Εικόνα 34: Περιστροφή

ROTATE-TO-ZERO(*points*)

- 1 $c \leftarrow \text{CENTROID}(\text{points})$ // computes (\bar{x}, \bar{y})
- 2 $\theta \leftarrow \text{ATAN}(c_y - \text{points}_{0_y}, c_x - \text{points}_{0_x})$ // for $-\pi \leq \theta \leq \pi$
- 3 $\text{newPoints} \leftarrow \text{ROTATE-BY}(\text{points}, -\theta)$
- 4 **return** *newPoints*

ROTATE-BY(*points*, θ)

- 1 $c \leftarrow \text{CENTROID}(\text{points})$
- 2 **foreach** point *p* in *points* **do**
- 3 $q_x \leftarrow (p_x - c_x) \cos \theta - (p_y - c_y) \sin \theta + c_x$
- 4 $q_y \leftarrow (p_x - c_x) \sin \theta + (p_y - c_y) \cos \theta + c_y$
- 5 $\text{APPEND}(\text{newPoints}, q)$
- 6 **return** *newPoints*

Βήμα 3

Μετά την περιστροφή το σχήμα πρέπει να αλλάξει κλίμακα. Η αλλαγή αυτή γίνεται μέσω ενός τετραγώνου αναφοράς. Αυτή η αλλαγή είναι ανομοιομορφη. Αφού γίνει η αλλαγή της κλίμακας μεταφράζουμε την χειρονομία σε ένα σημείο αναφοράς το οποίο επιλέγεται ώστε το κέντρο του σχήματος να βρίσκεται στο (0,0).

```

SCALE-TO-SQUARE(points, size)
1  B ← BOUNDING-BOX(points)
2  foreach point p in points do
3     $q_x \leftarrow p_x \times (size / B_{width})$ 
4     $q_y \leftarrow p_y \times (size / B_{height})$ 
5    APPEND(newPoints, q)
6  return newPoints

TRANSLATE-TO-ORIGIN(points)
1  c ← CENTROID(points)
2  foreach point p in points do
3     $q_x \leftarrow p_x - c_x$ 
4     $q_y \leftarrow p_y - c_y$ 
5    APPEND(newPoints, q)
6  return newPoints

```

Βήμα 4

Αφού γίνουν τα παραπάνω βήματα τότε πρέπει να βρεθεί το σωστό σχήμα. Για να γίνει αυτό συγκρίνεται το C με κάθε αποθηκευμένο σχήμα T_i και βρίσκεται η μέση απόσταση d_i ανάμεσα σε αυτά τα σημεία με την σχέση:

$$d_i = \frac{\sum_{k=1}^N \sqrt{(C[k]_x - T_i[k]_x)^2 + (C[k]_y - T_i[k]_y)^2}}{N}$$

Αυτή η σχέση δίνει το d_i το οποίο δίνει την απόσταση μεταξύ C και T_i . Το αποθηκευμένο σχήμα, το οποίο δίνει σαν αποτέλεσμα την μικρότερη απόσταση σύμφωνα με την παραπάνω σχέση, είναι το σχήμα που αναγνωρίζεται. Αυτή η ελάχιστη απόσταση μετατρέπεται σε μία άλλη παράμετρο score με σύνολο τιμών [0...1]. Η μεταβλητή size είναι το μήκος της μίας πλευράς του τετραγώνου αναφοράς.

$$score = 1 - \frac{d_i^*}{1/2 \sqrt{size^2 + size^2}}$$

Στο βήμα 2 υλοποιείται η περιστροφή, όμως αυτή η περιστροφή μπορεί να μην αρκεί για να βρεθεί το ελάχιστο μονοπάτι στο Ti. Για τον λόγο αυτό χρησιμοποιείται ο αλγόριθμος Golden Section Search. Ο golden section είναι μία τεχνική για την αναζήτηση του ελάχιστου ή του μέγιστου μίας αυστηρώς μονοκόρυφης συνάρτησης μέσω του διαδοχικού περιορισμού του εύρους τιμών αυτής, όπου τα ελάχιστα/μέγιστα είναι γνωστό ότι υπάρχουν. Είναι ένας αποδοτικός αλγόριθμος που βρίσκει το ελάχιστο, σε αυτή την περίπτωση, χρησιμοποιώντας για Golden Ratio $\phi=0.5(-1 + \sqrt{5})$. Παρακάτω φαίνεται ο αλγόριθμος όπου έχει χρησιμοποιηθεί $\theta=\pm 45^\circ$ και $\theta_\Delta=2^\circ$. Λόγω της επαναδειγματοληψίας μπορεί να θεωρηθεί ότι τα A και B περιέχουν τον ίδιο αριθμό σημείων άρα $|A|=|B|$.

```

RECOGNIZE(points, templates)
1   $b \leftarrow +\infty$ 
2  foreach template  $T$  in templates do
3     $d \leftarrow \text{DISTANCE-AT-BEST-ANGLE}(\textit{points}, T, -\theta, \theta, \theta_\Delta)$ 
4    if  $d < b$  then
5       $b \leftarrow d$ 
6       $T' \leftarrow T$ 
7     $\textit{score} \leftarrow 1 - b / 0.5\sqrt{(\textit{size}^2 + \textit{size}^2)}$ 
8  return  $\langle T', \textit{score} \rangle$ 

DISTANCE-AT-BEST-ANGLE(points,  $T$ ,  $\theta_a$ ,  $\theta_b$ ,  $\theta_\Delta$ )
1   $x_1 \leftarrow \phi\theta_a + (1 - \phi)\theta_b$ 
2   $f_1 \leftarrow \text{DISTANCE-AT-ANGLE}(\textit{points}, T, x_1)$ 
3   $x_2 \leftarrow (1 - \phi)\theta_a + \phi\theta_b$ 
4   $f_2 \leftarrow \text{DISTANCE-AT-ANGLE}(\textit{points}, T, x_2)$ 
5  while  $|\theta_b - \theta_a| > \theta_\Delta$  do
6    if  $f_1 < f_2$  then
7       $\theta_b \leftarrow x_2$ 
8       $x_2 \leftarrow x_1$ 
9       $f_2 \leftarrow f_1$ 
10    $x_1 \leftarrow \phi\theta_a + (1 - \phi)\theta_b$ 
11    $f_1 \leftarrow \text{DISTANCE-AT-ANGLE}(\textit{points}, T, x_1)$ 
12  else
13    $\theta_a \leftarrow x_1$ 
14    $x_1 \leftarrow x_2$ 
15    $f_1 \leftarrow f_2$ 
16    $x_2 \leftarrow (1 - \phi)\theta_a + \phi\theta_b$ 
17    $f_2 \leftarrow \text{DISTANCE-AT-ANGLE}(\textit{points}, T, x_2)$ 
18  return  $\text{MIN}(f_1, f_2)$ 

DISTANCE-AT-ANGLE(points,  $T$ ,  $\theta$ )
1   $\textit{newPoints} \leftarrow \text{ROTATE-BY}(\textit{points}, \theta)$ 
2   $d \leftarrow \text{PATH-DISTANCE}(\textit{newPoints}, T_{\textit{points}})$ 
3  return  $d$ 

PATH-DISTANCE( $A$ ,  $B$ )
1   $d \leftarrow 0$ 
2  for  $i$  from 0 to  $|A|$  step 1 do
3     $d \leftarrow d + \text{DISTANCE}(A_i, B_i)$ 
4  return  $d / |A|$ 

```


ΚΕΦΑΛΑΙΟ 5:Υλοποίηση

5.1 Καταγραφή δεδομένων

Για την υλοποίηση της εφαρμογής το βασικό δεδομένο που χρειάζεται για την αναγνώριση της χειρονομίας είναι η γωνιακή θέση του αντικείμενου. Το γυροσκόπιο υπολογίζει αυτήν την θέση μέσω της γωνιακής ταχύτητας που υπολογίζει ενώ το επιταχυνσιόμετρο υπολογίζει την g-force του αντικείμενου.

Πριν γίνει όμως η συλλογή δεδομένων πρέπει να γίνουν κάποιες επιλογές μέσω του Arduino IDE. Η συχνότητα του I2C ορίζεται 400kHz, το γυροσκόπιο ρυθμίζεται στα $\pm 250\text{deg/s}$, το επιταχυνσιόμετρο στα $\pm 2g$, απενεργοποιείται το FSYNC και το sleep mode. Επίσης για την σωστή λειτουργία του πρέπει να γίνει η ανάγνωση του I2C 0x75, το οποίο είναι γνωστό σαν MPU6050_RA_WHO_AM_I και πρέπει να επιστρέφει πάντα 0x68.

Η συλλογή δεδομένων γίνεται μέσω του mpu6050 με την χρήση του I2C. Το Arduino λαμβάνει από το mpu τις τιμές του επιταχυνσιόμετρου και του γυροσκοπίου μέσω του παρακάτω κώδικα:

```
accX = ((i2cData[0] << 8) | i2cData[1]);
```

```
accY = ((i2cData[2] << 8) | i2cData[3]);
```

```
accZ = ((i2cData[4] << 8) | i2cData[5]);
```

```
gyroX = ((i2cData[8] << 8) | i2cData[9]);
```

```
gyroY = ((i2cData[10] << 8) | i2cData[11]);
```

```
gyroZ = ((i2cData[12] << 8) | i2cData[13]);
```

Μπορούμε να υπολογίσουμε την γωνία (angle) του επιταχυνσιόμετρου:

```
accXangle = (atan2(accY, accZ) + PI) * RAD_TO_DEG;
```

```
accYangle = (atan2(accX, accZ) + PI) * RAD_TO_DEG;
```

Η atan2 έχει ένα εύρος τιμών $-\pi$ έως π . Η μετατροπή του εύρους σε 0 έως 2π γίνεται με την πρόσθεση π στην αρχική τιμή. Στη συνέχεια γίνεται η μετατροπή από rad σε μοίρες πολλαπλασιάζοντας με 57.295779513082320876798154814105 το οποίο γίνεται στο Arduino IDE με την RAD_TO_DEG.

Τα δεδομένα από το γυροσκόπιο αρχικά πρέπει να μετατραπούν σε μοίρες/sec. Για να γίνει αυτό πρέπει να διαιρεθούν με την ευαισθησία του γυροσκοπίου. Το datasheet εμπεριέχει τον παρακάτω πίνακα:

FS_SEL=0		131		LSB/(°/s)
FS_SEL=1		65.5		LSB/(°/s)
FS_SEL=2		32.8		LSB/(°/s)
FS_SEL=3		16.4		LSB/(°/s)

Εικόνα 35: Τιμές FS_SEL

Στον κώδικα χρησιμοποιήθηκε η default τιμή FS_SEL που είναι 0, άρα η ευαισθησία είναι 131. Για να μετατραπούν τα δεδομένα σε μοίρες πρέπει να είναι γνωστή η χρονική στιγμή από την τελευταία επανάληψη. Το Arduino έχει μία απλή εντολή για αυτήν την λειτουργία την: `micros()` η οποία υπολογίζει την διαφορά χρόνου (delta time) και με αυτήν υπολογίζεται η γωνιακή θέση του γυροσκοπίου. Τα παραπάνω γράφονται ως εξής:

```
double gyroXrate = (double)gyroX / 131.0;
```

```
double gyroYrate = -((double)gyroY / 131.0);
```

```
gyroXangle += gyroXrate * ((double)(micros() - timer) / 1000000);
```

5.2 Φιλτράρισμα

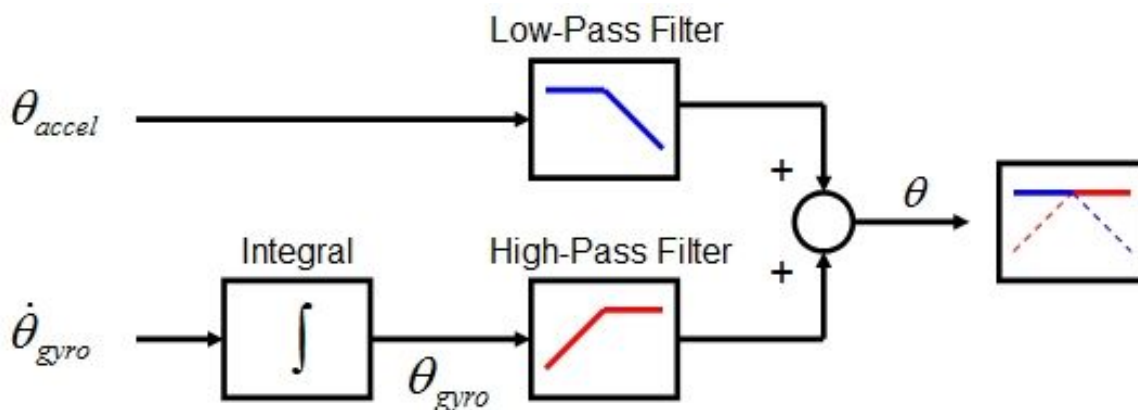
Η αναγνώριση των χειρονομιών μπορεί να γίνει με τα δεδομένα του επιταχυνσιόμετρου όμως όπως έχει αναφερθεί το επιταχυνσιόμετρο παρόλο που μπορούμε να το εμπιστευτούμε μακροπρόθεσμα, για μικρό διάστημα παρουσιάζει μεγάλη αστάθεια. Αντιθέτως το γυροσκόπιο παρουσιάζει αντίθετη συμπεριφορά. Για αυτό το λόγο ο συνδυασμός και των δύο αλλά και η χρήση ενός φίλτρου δίνει καλύτερα και ακριβή δεδομένα. Στην εφαρμογή χρησιμοποιήθηκε το Filter Kalman όμως για λόγους πληρότητας γίνεται αναφορά και σύγκριση και με ένα Complementary Filter.

5.2.1 Complementary Filter

Το Complementary Filter βραχυπρόθεσμα χρησιμοποιεί τα δεδομένα από το γυροσκόπιο επειδή είναι ακριβή και δεν υπόκεινται σε εξωτερικές δυνάμεις. Μακροπρόθεσμα χρησιμοποιεί τα δεδομένα του επιταχυνσιόμετρου. Στην απλή του μορφή το φίλτρο είναι ως εξής:

$$angle = 0.98 * (angle + gyrData * dt) + 0.02 * (accData)$$

Τα δεδομένα του γυροσκοπίου ενοποιούνται σε κάθε χρονικό βήμα με την παρούσα τιμή της γωνιακής τιμής. Στη συνέχεια συνδυάζονται με τα δεδομένα από το επιταχυνσιόμετρο τα οποία έχουν περάσει από ένα low pass filter. Στο low pass filter ελέγχεται αν η τιμή των δεδομένων του επιταχυνσιόμετρου είναι πολύ μικρή ή πολύ μεγάλη και στην περίπτωση αυτή αποτελεί θόρυβο που δεν λαμβάνεται υπόψη. Οι τιμές 0.98 και 0.02 πρέπει να αθροίζονται σε 1.00 και μπορούν να μεταβληθούν ώστε το φίλτρο να παρέχει το καλύτερο δυνατό αποτέλεσμα ανάλογα την εφαρμογή στην οποία χρησιμοποιείται.



Εικόνα 36: Complementary Filter

Τα παραπάνω σε μορφή κώδικα υλοποιούνται ως εξής:

```
compAngleX = (0.93 * (compAngleX + (gyroXrate * (double)(micros() - timer) / 1000000))) +  
(0.07 * accXangle);
```

```
compAngleY = (0.93 * (compAngleY + (gyroYrate * (double)(micros() - timer) / 1000000))) +  
(0.07 * accYangle);
```

5.2.2 Kalman Filter

Στη συνέχεια υλοποιείται το φίλτρο το οποίο χρησιμοποιήθηκε για την αναγνώριση. Γίνεται η μετατροπή των μαθηματικών εννοιών που έχουν περιγράψει ήδη, σε κώδικα. Οι εξισώσεις που είναι απαραίτητες αναπτύσσονται με μαθηματικό τρόπο.

Βήμα 1:

$$\begin{aligned}
 \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F} \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B} \dot{\theta}_k \\
 \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} &= \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k-1|k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_k \\
 &= \begin{bmatrix} \theta - \dot{\theta}_b \Delta t \\ \dot{\theta}_b \end{bmatrix}_{k-1|k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_k \\
 &= \begin{bmatrix} \theta - \dot{\theta}_b \Delta t + \dot{\theta} \Delta t \\ \dot{\theta}_b \end{bmatrix} \\
 &= \begin{bmatrix} \theta + \Delta t (\dot{\theta} - \dot{\theta}_b) \\ \dot{\theta}_b \end{bmatrix}
 \end{aligned}$$

Η a priori εκτίμηση της γωνίας $\hat{\theta}_{k|k-1}$ είναι ίση με την εκτίμηση της προηγούμενης κατάστασης συν την μέτρηση χωρίς το σφάλμα επί το delta time Δt . Επειδή δεν μπορούμε απευθείας να υπολογίσουμε το σφάλμα, η εκτίμηση του σφάλματος της a priori κατάστασης είναι ίση με την προηγούμενη. Αυτό γράφεται ως εξής:

```
rate = newRate - bias;
angle += dt * rate;
```

Βήμα 2:

$$\begin{aligned}
 \mathbf{P}_{k|k-1} &= \mathbf{F} \mathbf{P}_{k-1|k-1} \mathbf{F}^T + \mathbf{Q}_k \\
 \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} &= \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k-1|k-1} \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} + \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t \\
 &= \begin{bmatrix} P_{00} - \Delta t P_{10} & P_{01} - \Delta t P_{11} \\ P_{10} & P_{11} \end{bmatrix}_{k-1|k-1} \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} + \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t \\
 &= \begin{bmatrix} P_{00} - \Delta t P_{10} - \Delta t (P_{01} - \Delta t P_{11}) & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11} \end{bmatrix}_{k-1|k-1} + \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t \\
 &= \begin{bmatrix} P_{00} - \Delta t P_{10} - \Delta t (P_{01} - \Delta t P_{11}) + Q_\theta \Delta t & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11} + Q_{\dot{\theta}_b} \Delta t \end{bmatrix}
 \end{aligned}$$

Οι παραπάνω εξισώσεις υλοποιούνται ως εξής:

```
P[0][0] += dt * (dt*P[1][1] - P[0][1] - P[1][0] + Q_angle);
P[0][1] -= dt * P[1][1];
P[1][0] -= dt * P[1][1];
P[1][1] += Q_gyroBias * dt;
```

Βήμα 3:

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}\hat{x}_{k|k-1} \\ &= \mathbf{z}_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} \\ &= \mathbf{z}_k - \theta_{k|k-1}\end{aligned}$$

```
y = newAngle - angle;
```

Βήμα 4:

$$\begin{aligned}\mathbf{S}_k &= \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \\ &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathbf{R} \\ &= P_{00k|k-1} + \mathbf{R} \\ &= P_{00k|k-1} + var(v)\end{aligned}$$

```
S = P[0][0] + R_angle;
```

Βήμα 5:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \\ \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k &= \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{S}_k^{-1} \\ &= \begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1} \mathbf{S}_k^{-1}\end{aligned}$$

$$= \frac{\begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1}}{S_k}$$

$$K[0] = P[0][0] / S;$$

$$K[1] = P[1][0] / S;$$

Βήμα 6:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k} = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \tilde{\mathbf{y}}_k$$

$$= \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \tilde{\mathbf{y}} \\ K_1 \tilde{\mathbf{y}} \end{bmatrix}_k$$

$$\text{angle} += K[0] * y;$$

$$\text{bias} += K[1] * y;$$

Βήμα 7:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1}$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1}$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 & 0 \\ K_1 & 0 \end{bmatrix}_k \right) \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1}$$

$$= \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_0 P_{00} & K_0 P_{01} \\ K_1 P_{00} & K_1 P_{01} \end{bmatrix}$$

Ο πίνακας λαθών μειώνεται ξανά διότι έχει μειωθεί το λάθος της κατάστασης εκτίμησης

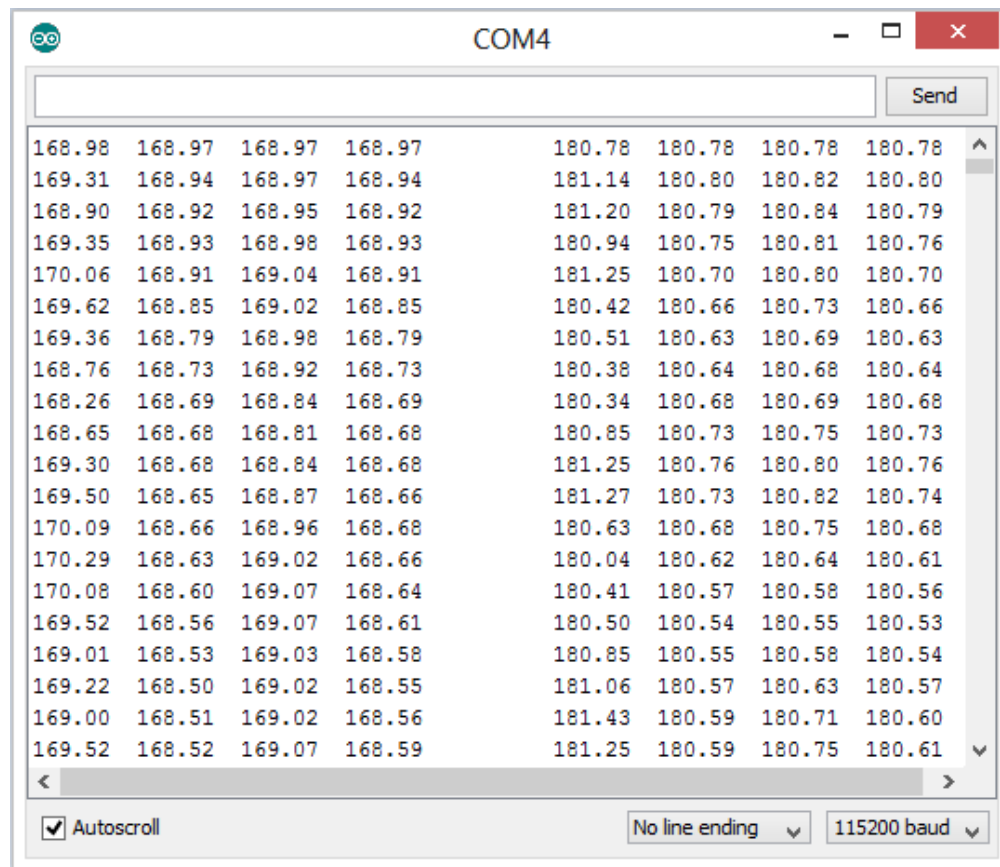

```

P[0][0] -= K[0] * P[0][0];
P[0][1] -= K[0] * P[0][1];
P[1][0] -= K[1] * P[0][0];
P[1][1] -= K[1] * P[0][1];

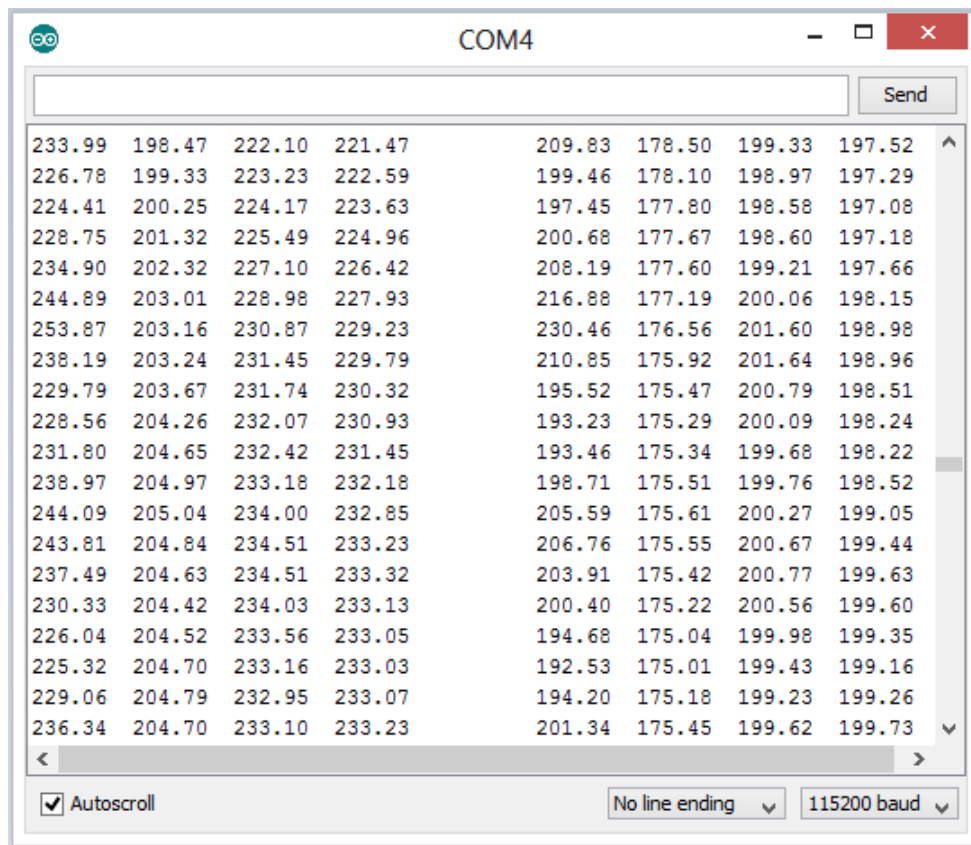
```

5.2.3 Επιλογή δεδομένων

Με την παραπάνω διαδικασία μπορούμε να αναγνωρίσουμε την κίνηση και το είδος της χειρονομίας με τέσσερις τρόπους. Με τα δεδομένα του επιταχυνσιόμετρου, με του γυροσκοπίου, με τα αποτελέσματα της επεξεργασίας του Complementary Filter και του Kalman Filter. Στις παρακάτω εικόνες φαίνονται αυτά τα δεδομένα μέσα από το Serial Monitor του Arduino Ide. Στην πρώτη εικόνα φαίνονται τα δεδομένα από μία αρχική χρονική στιγμή ενώ στην δεύτερη από μία μετέπειτα κατά την οποία η συσκευή έχει μετατοπιστεί. Οι στήλες αντιστοιχούν στα δεδομένα από το επιταχυνσιόμετρο, το γυροσκόπιο, το Complementary Filter και το Kalman Filter αντίστοιχα. Οι πρώτες τέσσερις αναφέρονται στο διάνυσμα x και οι δεύτερες στο y .

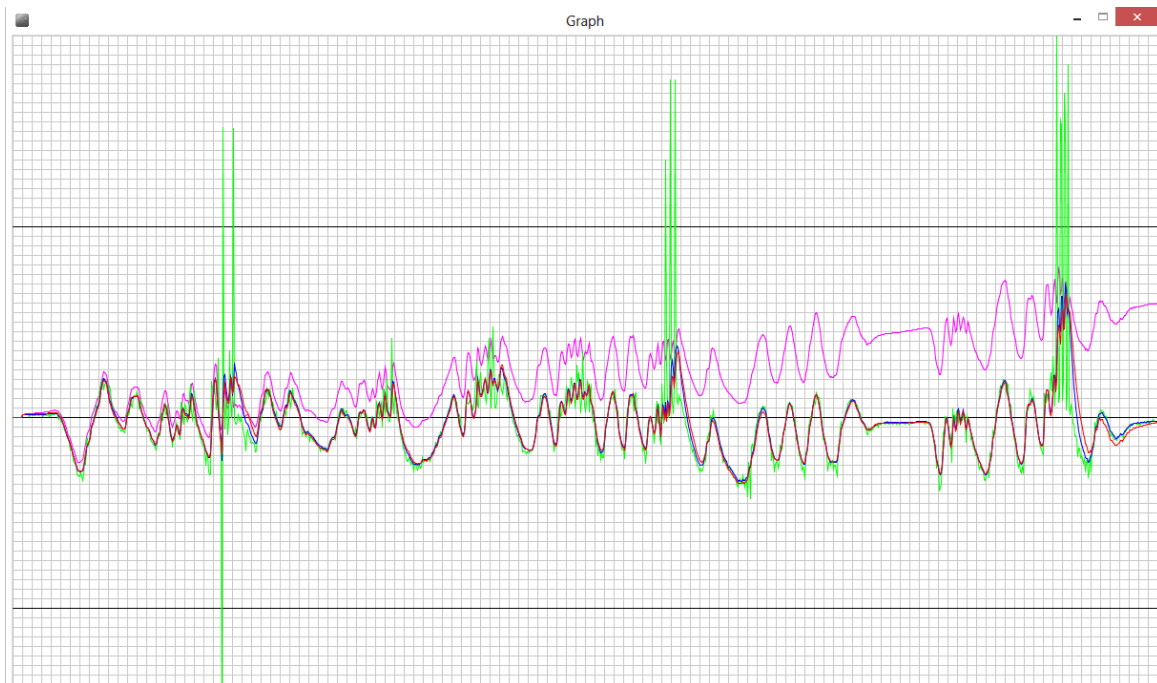


Εικόνα 37: Δεδομένα Arduino αρχικής στιγμής



Εικόνα 38: Δεδομένα Arduino τυχαίας στιγμής

Τα δεδομένα αυτά μεταβάλλονται άλλοτε πιο σταθερά και άλλοτε πιο γρήγορα ανάλογα με το είδος της κίνησης και με την κατεύθυνση της. Όπως έχει αναφερθεί τα δεδομένα του επιταχυνσιόμετρου και του γυροσκοπίου από μόνα τους δεν είναι αξιόπιστα. Για αυτό το λόγο πρέπει να επιλεγεί η χρήση ενός φίλτρου. Παρατηρώντας τα δεδομένα δεν είναι ξεκάθαρο ποιος τρόπος πρέπει να επιλεγεί. Για αυτόν τον λόγο μέσω του Processing μπορεί να γίνει η γραφική παράσταση των δεδομένων αυτών και να γίνει πιο φανερή η διαδικασία που πρέπει να ακολουθηθεί.



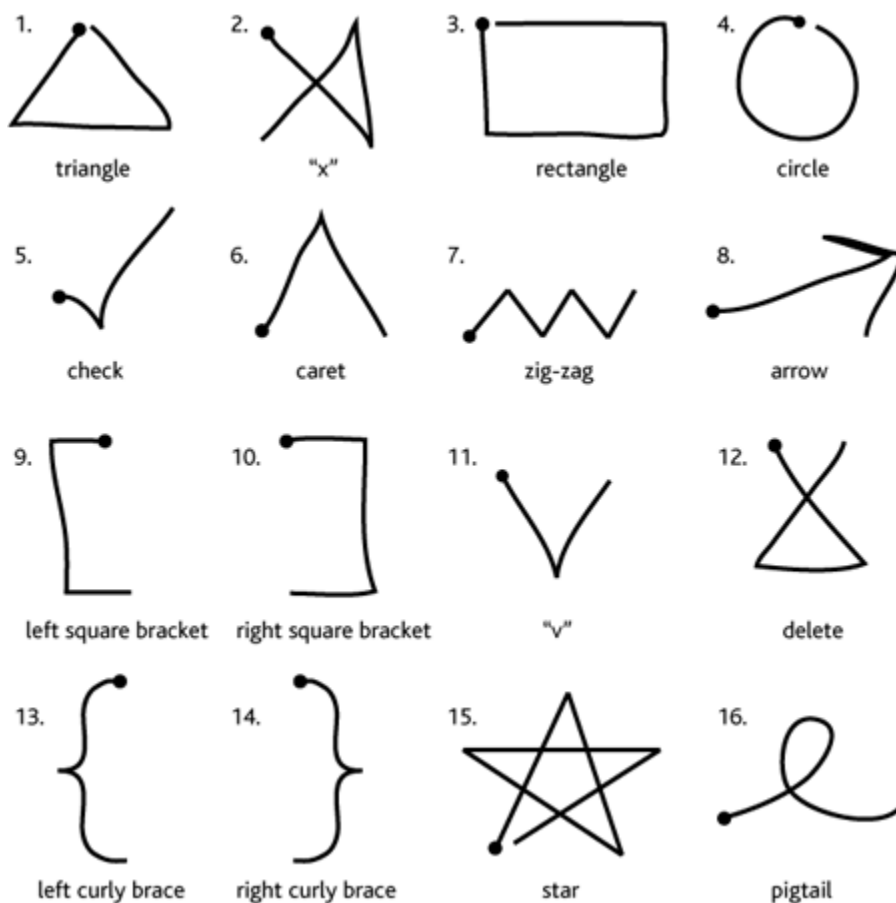
Εικόνα 39: Γραφική παράσταση με όλους τους τρόπους συλλογής δεδομένων

Στο παραπάνω γράφημα φαίνονται τα δεδομένα όπως αυτά καταγράφονται από την συσκευή. Το χρώμα μωβ αντιπροσωπεύει τα δεδομένα του γυροσκοπίου και όπως φαίνεται μακροπρόθεσμα χάνει την αξιοπιστία του. Με πράσινο χρώμα είναι τα δεδομένα του επιταχυνσιόμετρου τα οποία σε απότομες μεταβολές της συσκευής επίσης δεν δίνουν αξιόπιστες τιμές. Με μπλε και με κόκκινο χρώμα παρουσιάζονται τα φίλτρα που χρησιμοποιήθηκαν. Με μπλε το Complementary και με κόκκινο το Kalman. Τα δύο φίλτρα δίνουν αξιόπιστες τιμές και η λειτουργία τους είναι σωστή. Οι τιμές που δίνουν είναι σχεδόν ίδιες. Όμως το Kalman είναι λίγο πιο ακριβές στις τιμές του ειδικότερα σε στιγμές απότομης μεταβολής της συσκευής. Για αυτό τον λόγο η αναγνώριση της χειρονομίας θα πραγματοποιηθεί με την χρήση αυτών των δεδομένων δηλαδή του Kalman Filter.

5.3 Αναγνώριση χειρονομιών

Αμέσως μετά το φιλτράρισμα υλοποιείται ο αλγόριθμος αναγνώρισης χειρονομιών ο οποίος βασίζεται στα στάδια που έχουν περιγράψει. Ο κώδικας βρίσκεται συγκεντρωμένος στην εξής τοποθεσία <https://www.dropbox.com/s/kru2bh1ac2zc83s/Arduino.rar?dl=0>. Για την λειτουργία του πρέπει να υπάρχουν αποθηκευμένα ορισμένα templates. Ένα από αυτά φαίνεται παρακάτω, συνολικά είναι 16 ενώ μπορεί να προστεθούν και άλλα ανάλογα τις επιθυμίες και τις ανάγκες αναγνώρισης.

```
one.add("triangle",newInteger[]{ 137,139,135,141,133,144,132,146,130,149,128,151,126,155,123,160,120,166,116,171,112,177,107,183,102,188,100,191,95,195,90,199,86,203,82,206,80,209,75,213,73,213,70,216,67,219,64,221,61,223,60,225,62,226,65,225,67,226,74,226,77,227,85,229,91,230,99,231,108,232,116,233,125,233,134,234,145,233,153,232,160,233,170,234,177,235,179,236,186,237,193,238,198,239,200,237,202,239,204,238,206,234,205,230,202,222,197,216,192,207,186,198,179,189,174,183,170,178,164,171,161,168,154,160,148,155,143,150,138,148,136,148} );
```



Εικόνα 40: Σχήματα αναγνώρισης

Με βάση το παραπάνω σχήμα και μετά από ορισμένες κινήσεις τις συσκευής μπορούν να αναγνωριστούν οι χειρονομίες. Για καλύτερο οπτικό αποτέλεσμα με την βοήθεια του Processing σχεδιάζεται το σχήμα που πραγματοποιείται. Ακολουθούν ορισμένα παραδείγματα:



Εικόνα 41: Αναγνώριση circle



Εικόνα 44: Αναγνώριση v



Εικόνα 45: Αναγνώριση pigtail



Εικόνα 46: Αναγνώριση caret

Τα σχήματα που εμφανίζονται στην οθόνη του processing αναγνωρίζονται με τον αλγόριθμο που χρησιμοποιήθηκε με επιτυχία. Εμφανίζεται στην οθόνη η πληροφορία ότι κάποια χειρονομία ανιχνεύτηκε και εμφανίζει το όνομα της. Παράλληλα εμφανίζεται το ποσοστό κατά το οποίο η χειρονομία που πραγματοποιήθηκε ταιριάζει με το template που είναι αποθηκευμένο. Το ποσοστό αυτό είναι πάνω του 80% ενώ μικρότερες τιμές του δεν εμφανίζονται στην οθόνη αφού η χειρονομία δεν μπορεί να ανιχνευτεί. Εκτός από τα παραπάνω παραδείγματα μπορούν να αναγνωριστούν και τα υπόλοιπα σχήματα τα οποία βρίσκονται αποθηκευμένα.

ΚΕΦΑΛΑΙΟ 6: Επίλογος

6.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε το σχεδιασμό και την ανάπτυξη μίας συσκευής αναγνώρισης κινήσεων και σχημάτων με την βοήθεια της τεχνολογίας Arduino. Χωρίστηκε σε δύο φάσεις. Η πρώτη, αφορούσε την συλλογή δεδομένων από το περιβάλλον με την βοήθεια δύο οργάνων, ενός επιταχυνσιόμετρου και ενός γυροσκοπίου τριών διαστάσεων ενσωματωμένα στο MPU6050 και το φιλτράρισμα και την επεξεργασία αυτών των δεδομένων για την καλύτερη απόδοση της συσκευής. Η δεύτερη φάση αφορούσε την αναγνώριση με βάση ενός αλγορίθμου και την γραφική αναπαράσταση της αναγνώρισης του σωστού σχήματος.

Ο χρήστης ενεργοποιεί την συσκευή, την μετακινεί και αναλόγως του σχήματος το ανάλογο αποτέλεσμα εμφανίζεται στην οθόνη. Η λειτουργία της συσκευής είναι αρκετά εύκολη και προσίτη χωρίς δύσκολους μαθηματικούς υπολογισμούς ενώ ο αλγόριθμος είναι γρήγορος και αποδοτικός. Τα αποτελέσματα που εμφανίζονται είναι σε μεγάλο βαθμό σωστά που κάνει την συσκευή αξιόπιστη. Τα σφάλματα και οι αποκλίσεις είναι ελάχιστα χάρη στο φίλτρο που χρησιμοποιήθηκε.

Όμως ο αλγόριθμος έχει κάποιους περιορισμούς. Ο 1^{ος} αλγόριθμος όπως έχει αναφερθεί συγκρίνει τα υποψήφια σημεία της χειρονομίας με μία σειρά από αποθηκευμένα templates. Η διαδικασία που ακολουθείται για να υλοποιηθεί όμως δεν μπορεί να διαχωρίσει συγκεκριμένα σχήματα. Για παράδειγμα ο διαχωρισμός τετραγώνων και παραλληλόγραμμων, κύκλων και ελλείψεων δεν είναι δυνατός χωρίς την αλλαγή του κώδικα. Επίσης λόγω της αλλαγής κλίμακας ο διαχωρισμός μικρών σχημάτων και μεγάλων δεν μπορεί να γίνει. Τέλος ο αλγόριθμος είναι ανεξάρτητος του χρόνου της κίνησης άρα η χειρονομία δεν μπορεί να διαφοροποιηθεί με βάση την ταχύτητα της κίνησης.

6.2 Μελλοντικές επεκτάσεις

Η αναγνώριση χειρονομιών βρίσκει εφαρμογή ήδη σε πολλές συσκευές όπως κινητά, tablet και σε άλλες ηλεκτρονικές συσκευές. Στο μέλλον είναι σίγουρο ότι αυτός ο τομέας της αλληλεπίδρασης ανθρώπου υπολογιστή θα μπει σε μεγάλο βαθμό στην καθημερινή ζωή των ανθρώπων. Ειδικότερα με την αύξηση και την ανάπτυξη της τεχνολογίας νέοι τρόποι όπως το Internet of Things θα δώσουν την δυνατότητα στον άνθρωπο να ελέγχει όλο και περισσότερους τομείς της ζωής του μόνο με μία κίνηση των χεριών του. Στα παραπάνω το μόνο πρόβλημα που εισάγεται είναι το κόστος. Όμως η συσκευή που κατασκευάστηκε και η τεχνολογία από την οποία έγινε δίνει την δυνατότητα επέκτασης και σχεδίασης πολλών νέων με καινούργιες δυνατότητες εύκολα και χωρίς κόστος. Αυτό γίνεται διότι το Arduino ανήκει στην κατηγορία

του Open Hardware. Με αυτόν τον τρόπο και με την συνεχή ανάπτυξη του Open Hardware προβλήματα όπως είναι οι άδειες κατασκευής παύουν να υπάρχουν. Τέλος ο αλγόριθμος μπορεί να επεκταθεί σε μεγάλο βαθμό ώστε να ξεπεραστούν οι περιορισμοί του και να μπορεί να αναγνωρίζει και να ξεχωρίζει ακόμα περισσότερες χειρονομίες σχημάτων που μοιάζουν μεταξύ τους.

Βιβλιογραφία

- [1] Αβούρης Ν. (2000). “Εισαγωγή στην Επικοινωνία Ανθρώπου-Υπολογιστή” Εκδόσεις Δίαυλος Αθήνα
- [2] Κουτσαμπάσης, Π. (2011) Αλληλεπίδρασης Ανθρώπου – Υπολογιστή Ανθρώπου-Υπολογιστή: Μέθοδοι, Αρχές και Παραδείγματα, Κλειδάριθμος
- [3] Manresa, C., Varona, J., Mas, R. and Perales, F. J. (2005) Hand Tracking and Gesture Recognition for Human-Computer Interaction, In Electronic Letters on Computer Vision and Image Analysis
- [4] Gan, L., (2011). Real-time Immersive human-computer interaction based on tracking and recognition of dynamic hand gestures.
- [5] Sturman, D.J., Zeltzer, D. (1993). A design method for "whole-hand human computer interaction". ACM transactions on information systems
- [6] Quam, D.L. (1990). Gesture Recognition With a DataGlove, Proc. IEEE National Aerospace and Electronics
- [7] Starner, T.E., Pentland, A. (1995). Real-Time American Sign Language Recognition from Video using Hidden Markov Models
- [8] Arduino playground, <http://www.arduino.cc/>, Σεπτέμβριος 2014
- [9] Arduino Project. <http://www.arduino.cc/>, Σεπτέμβριος 2014
- [10] <http://arduino.cc/en/Tutorial/HomePage>, Σεπτέμβριος 2014
- [11] <http://arduino.cc/en/Main/arduinoBoardUno>, Σεπτέμβριος 2014
- [12] Arduino IDE, <http://arduino.cc/en/Guide/Environment>, Σεπτέμβριος 2014
- [13] Ιωάννα Παπαγεωργίου , Αξιοποίηση τεχνολογιών internet of things για την διαχείριση οδικών αξόνων
- [14] Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. Proceedings of the ACM Symposium on User Interface Software and Technology
- [15] Tappert, C.C. (1982). Cursive script recognition by elastic matching. IBM Journal of Research and Development.