

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ασύρματη καταγραφή κίνησης χεριού

Νικόλαος Γ. Κωνσταντακόπουλος

**Εισηγητές: Δρ. Ιωάννης Έλληνας, Καθηγητής
Στέλιος Βουτσινάς, Καθηγητής**

**ΑΘΗΝΑ
ΦΕΒΡΟΥΑΡΙΟΣ 2018**

Ασύρματη καταγραφή κίνησης χεριού

Ασύρματη καταγραφή κίνησης χεριού

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ασύρματη καταγραφή κίνησης χεριού

**Νικόλαος Γ. Κωνσταντακόπουλος
Α.Μ. 40738**

Εισηγητές:

**Δρ. Ιωάννης Έλληνας, Καθηγητής
Στέλιος Βουτσινάς, Καθηγητής**

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης 27/3/2018

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Νικόλαος Κωνσταντακόπουλος, του Γεωργίου, με αριθμό μητρώου 40738 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ασύρματη καταγραφή κίνησης χεριού

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των σύγχρονων τεχνολογιών Arduino και την κατασκευή κυκλωμάτων, με σκοπό την ασύρματη καταγραφή της κίνησης του χεριού και την απεικόνισής της. Την προσπάθειά μου αυτή υποστήριξαν οι επιβλέποντες καθηγητές μου Ιωάννης Έλληνας και Στέλιος Βουτσινάς, τους οποίους θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω τον κ. Κωνσταντίνο Ζαφειρούλη για τις πολύτιμες συμβουλές του, καθώς και την οικογένεια μου για την τεχνική και ψυχολογική υποστήριξη.

Ασύρματη καταγραφή κίνησης χεριού

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ασύρματη καταγραφή της κίνησης του χεριού με σύγχρονες τεχνολογίες, όπως αυτή του Arduino, καθώς και με την απεικόνιση αυτής της κίνησης σε πραγματικό χρόνο. Αρχικά, αναφέρονται κάποιες εισαγωγικές έννοιες γύρω από την τεχνολογία Arduino, ενώ στη συνέχεια γίνεται παρουσίαση των επιμέρους εξαρτημάτων που χρησιμοποιήθηκαν για την κατασκευή. Εφόσον ασχολούμαστε με κίνηση, ήταν χρήσιμο να αναλυθεί ο τρόπος λειτουργίας των αισθητήρων κίνησης (γυροσκόπιο-επιταχυνσιόμετρο), καθώς και οι δυνατότητες του αισθητήρα που χρησιμοποιήθηκε. Έπειτα, επισημαίνεται ο τρόπος επίτευξης της ασύρματης επικοινωνίας. Τέλος, αναλύεται το λογισμικό, το οποίο πραγμάτωσε την απεικόνιση (Unity), καθώς και ο τρόπος προσαρμογής των μετρήσεων των αισθητήρων στο 3d μοντέλο, το οποίο επιλέχθηκε.

ABSTRACT

This diploma thesis concerns the wireless hand's motion capture with modern technologies such as Arduino, as well as the visualization of this movement in real time. Initially, some introductory concepts are mentioned around Arduino technology, followed by the presentation of each sub-part of the circuit. Since we are dealing with motion capture, it was useful to analyze how motion sensors work (gyroscope-accelerometer) as well as the potential of the sensor used. Afterward, the way to achieve wireless communication is highlighted. Finally, I analyze the software that made the visualization (Unity), as well as the way that I adapt the sensor measurements to the 3D model, which was chosen.

ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΗ	15
1.1	Περιγραφή του αντικειμένου της πτυχιακής εργασίας	15
1.2	Ιστορική αναδρομή	16
1.3	Ανασκόπηση της πτυχιακής εργασίας	19
2.	ΤΕΧΝΟΛΟΓΙΑ ARDUINO	21
2.1	Εισαγωγή στο Arduino	21
2.2	Μικροελεγκτής Arduino	22
2.3	Εισαγωγή στα βασικά μοντέλα Arduino	23
2.4	Λογισμικό Arduino	28
2.5	Πλεονεκτήματα Arduino	30
3.	ΚΑΤΑΣΚΕΥΗ	31
3.1	Τεχνολογίες που χρησιμοποιήθηκαν	31
3.2	Μητρική πλακέτα κυκλώματος	32
3.2.1	Adafruit Pro trinket	32
3.2.2	Πλεονεκτήματα Adafruit Pro trinket	35
3.2.3	Adafruit Pro trinket liion/lipoly backpack add-on	35
3.3	Καταγραφή κίνησης	37
3.3.1	Εισαγωγή στην καταγραφή της κίνησης	37
3.3.2	Γυροσκόπιο (gyroscope)	38
3.3.3	Επιταχυνσιόμετρο (accelerometer)	39
3.3.4	Γωνίες Euler	41
3.3.5	Τετραδόνια (quaternions)	43
3.3.6	Σύστημα αναφοράς θέσης στίγματος (AHRS)	46
3.3.7	Αδρανειακοί αισθητήρες (IMU)	46
3.3.8	Αδρανειακός αισθητήρας κατασκευής	47
3.3.8.1	MPU6050	47
3.3.8.2	Δίαυλος I2C	49
3.3.8.3	DMP	51
3.3.8.4	Πλακέτα GY-521	51

3.3.8.5 Βαθμονόμηση αισθητήρα κίνησης	52
3.4 Ασύρματη επικοινωνία	54
3.4.1 HC-05	54
3.4.2 Επικοινωνία συσκευής με σειριακή πόρτα	56
3.5 Παρουσίαση κατασκευής.....	58
3.5.1 Εισαγωγή στην κατασκευή	58
3.5.2 Συνδεσμολογία	59
3.5.3 Ολοκλήρωση κατασκευής	62
3.5.4 Εικόνες κατασκευής	63
3.5.5 Επεξήγηση κώδικα κατασκευής	64
4. ΑΠΕΙΚΟΝΙΣΗ ΚΙΝΗΣΗΣ ΧΕΡΙΟΥ ΣΕ 3D ΜΟΝΤΕΛΟ.....	68
4.1 Μοντελοποίηση 3D	68
4.2 Σχεδιοκίνηση τριών διαστάσεων (3D animation)	68
4.3 Λογισμικό Unity	70
4.3.1 Εισαγωγή στο Unity	70
4.3.2 Πλεονεκτήματα Unity	71
4.3.3 Έτοιμα μοντέλα 3D/ Asset store	71
4.3.4 Βασικές έννοιες και λειτουργίες περιβάλλοντος Unity	72
4.4 Υλοποίηση απεικόνισης	75
4.4.1 Επιλογή 3D μοντέλου	75
4.4.2 Ανάλυση επιμέρους κομματιών 3D μοντέλου	77
4.4.3 Περιστροφή (Rotation) μοντέλου στο χώρο	80
4.4.4 Επεξήγηση κώδικα απεικόνισης	80
5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	83
5.1 Σύνοψη της πτυχιακής εργασίας	83
5.2 Προοπτικές	83
6. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΙΣΤΟΣΕΛΙΔΕΣ	85

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: Πρώτη απόδοση κίνησης χαρακτήρα μέσω ροτοσκοπίου	16
Εικόνα 2.1: Λογότυπο Arduino	22
Εικόνα 2.2: ATmega328.....	22
Εικόνα 2.3: Arduino Uno R3	24
Εικόνα 2.4: Arduino Leonardo	25
Εικόνα 2.5: Arduino Mega.....	25
Εικόνα 2.6: Arduino Ethernet	26
Εικόνα 2.7: Arduino Yun	27
Εικόνα 2.8: Arduino Mini	27
Εικόνα 2.9: Arduino Lilypad	28
Εικόνα 2.10: Arduino sketch	29
Εικόνα 3.1: Adafruit Pro Trinket 5V	32
Εικόνα 3.2: Διάγραμμα ακίδων (pinout) Adafruit Pro Trinket 5V	33
Εικόνα 3.3: Adafruit Pro Trinket Liion/Lipoly Backpack Add-on	35
Εικόνα 3.4: Συνδεσμολογία Adafruit Pro Trinket Liion/Lipoly Backpack Add-on	36
Εικόνα 3.5: Γυροσκόπιο	38
Εικόνα 3.6: Δύναμη Coriolis	39
Εικόνα 3.7: Επιταχυνσιόμετρο	40
Εικόνα 3.8: Yaw-Pitch-Roll	41
Εικόνα 3.9: Γωνίες Euler	42
Εικόνα 3.10: Περιστροφή μοναδιαίων τετραδονίων στον τετραδιάστατο χώρο..	44
Εικόνα 3.11: MPU6050	47
Εικόνα 3.12: Block diagram του MPU6050	49
Εικόνα 3.13: Δίαυλος I2C	50
Εικόνα 3.14: GY-521	51
Εικόνα 3.15: Σύνδεση Arduino Uno με MPU6050	53
Εικόνα 3.16: Επιμέρους τμήματα πλακέτας HC-05	54
Εικόνα 3.17: Ακροδέκτες HC-05	55
Εικόνα 3.18: Ιδιότητες συσκευής HC-05 στον πίνακα ελέγχου των Windows ...	57
Εικόνα 3.19 Ιδιότητες σειριακής πόρτας στη διαχείριση συσκευών των Windows	57

Εικόνα 3.20: Σύνδεση κεντρικής πλακέτας-βοηθητικής-μπαταρίας-διακόπτη	59
Εικόνα 3.21: Κύκλωμα με προσθήκη της bluetooth συσκευής HC-05	60
Εικόνα 3.22: Συνδεσμολογία τελικού κυκλώματος	61
Εικόνα 3.23: Block diagram τελικού κυκλώματος	61
Εικόνα 3.24: Συνδέσεις εξαρτημάτων και διάτρηση κουτιού	63
Εικόνα 3.25: Τοποθέτηση κυκλώματος στο κουτί και τελική μορφή παρουσίασης	63
Εικόνα 3.26: Offset αισθητήρων μου κατά την βαθμονόμηση (calibration)	66
Εικόνα 4.1: Logo λογισμικού Unity	70
Εικόνα 4.2: Παράθυρα (windows) περιβάλλοντος Unity	73
Εικόνα 4.3: Διαδικασία rigging τρισδιάστατου μοντέλου χεριού	76
Εικόνα 4.4: Objects πακέτου Fps Arm Pack	77
Εικόνα 4.5: Σημεία αρθρώσεων καταγραφής κίνησης	79

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1: Χαρακτηριστικά ATmega328	23
--	----

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

AHRS Attitude and Heading Reference System

CPU Central Processing Unit

DOF Degrees Of Freedom

IMU Inertial Measurement Unit

I2C Inter-integrated Circuit

LiPo Lithium Polymer (battery)

mAh Milliampere hour

MEMES Microelectromechanical System

MPU Motion Processing Unit

USB Universal Serial Bus

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια ιστορική αναδρομή γύρω από τις μεθόδους καταγραφής και απεικόνισης της κίνησης σε πραγματικό χρόνο.

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

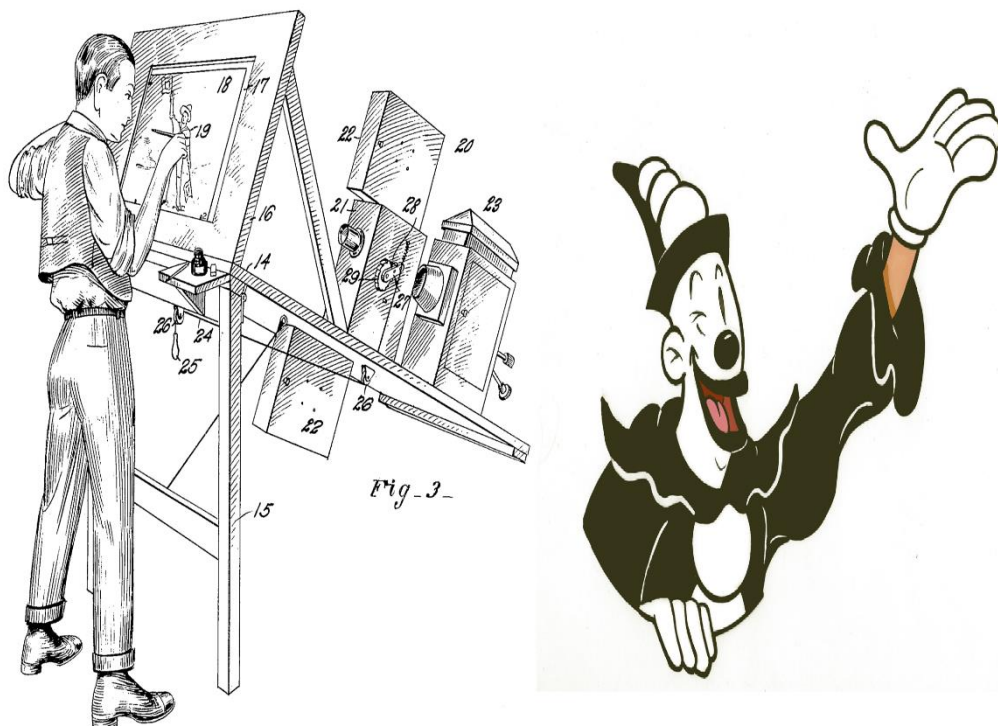
Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ασύρματη καταγραφή της κίνησης του χεριού. Πρωταρχικός στόχος ήταν η κατασκευή δύο πανομοιότυπων κυκλωμάτων, τα οποία θα τοποθετούνταν στο δεξί χέρι, το ένα μεταξύ ώμου - αγκώνα και το άλλο μεταξύ αγκώνα – καρπού, με όσο το δυνατόν μικρότερες διαστάσεις για ευελιξία κινήσεων. Τα δύο κυκλώματα θα έπρεπε ασύρματα να μεταδίδουν τις μετρήσεις των αισθητήρων κίνησης, ώστε να είναι δυνατή η απεικόνιση της ολοκληρωμένης κίνησης ενός χεριού από τον ώμο μέχρι τον καρπό. Έτσι λοιπόν, για κάθε κύκλωμα επέλεξα μια κεντρική πλακέτα Arduino ανοιχτού κώδικα, την Adafruit Pro Trinket, η οποία προγραμματίζεται εύκολα με mini USB, ένα bluetooth module για ασύρματη επικοινωνία, το HC-05, καθώς και έναν αισθητήρα κίνησης με γυροσκόπιο και επιταχυνσιόμετρο, τον MPU6050. Κάθε κύκλωμα τροφοδοτείται από μία μπαταρία (Li-Po 3.7V-250mAh), την οποία συνέδεσα με ένα module που προσφέρει η Adafruit για την κεντρική μας πλακέτα, ώστε να μπορεί να φορτίζεται η Li-Po μπαταρία μας μέσω του usb.

Επιλέχθηκε η καταγραφή της κίνησης του δεξιού χεριού από των ώμο μέχρι τον καρπό. Η απεικόνιση πραγματοποιείται σε ένα 3d μοντέλο ενός ολόκληρου δεξιού χεριού (από ώμο μέχρι άκρα) . Ο λόγος που επέλεξα το 3d μοντέλο, να είναι μόνο ένα χέρι και όχι ολόκληρος ο άνθρωπος, είναι διότι ήθελα να εστιάσω στον τρόπο περιστροφής του χεριού. Τέλος είναι σημαντικό να επισημανθεί ότι η απεικόνιση επαφίεται στην περιστροφική κίνηση του χεριού στους 3 άξονες (x,y,z) και όχι στην μετακίνηση του μέσα στο χώρο.

1.2 Ιστορική αναδρομή

Η καταγραφή κίνησης (motion capture ή mo-cap) για χρήση σε ψηφιακά μοντέλα δισδιάστατων ή τρισδιάστατων χαρακτήρων είναι μια σχετικά πρόσφατα αναπτυσσόμενη τεχνολογία, η οποία με τη μορφή που τη συναντάμε σήμερα ξεκινάει στα τέλη του '70 και αρχίζει να διαδίδεται μια δεκαετία αργότερα. Η διαδικασία περιλαμβάνει την χαρτογράφηση της κίνησης του ανθρώπινου σώματος πάνω στην κίνηση ενός χαρακτήρα του υπολογιστή.

Τα πρώιμα στάδια της σύλληψης της κίνησης έχουν τις ρίζες τους το 1915, όπου ο Max Fleisher με τη βοήθεια ενός ροτοσκοπίου (μέσω αντιγραφής μιας ρεαλιστικής κίνησης από ένα βίντεο σε χαρακτήρα κινούμενων σχεδίων) κατάφερε να αποδώσει την κίνηση από βιντεοσκόπηση του αδερφού του David Fleisher, στον χαρακτήρα κινούμενων σχεδίων της εποχής koko the clown. Το 1937, ο Walt Disney χρησιμοποίησε την ίδια τεχνική στην ταινία η χιονάτη και οι επτά νάνοι για τη μέγιστη δυνατή απόδοση ρεαλισμού κίνησης της χιονάτης.



Εικόνα 1.1: Πρώτη απόδοση κίνησης χαρακτήρα μέσω ροτοσκοπίου (1915)

Στα τέλη του 1970, γίνεται δυνατή η απόδοση κίνησης σε χαρακτήρα μέσω υπολογιστή και έτσι από τότε και έπειτα έχουμε την απόδοση της βιντεοσκοπημένης κίνησης σε ψηφιακούς χαρακτήρες. Με την αναπτυσσόμενη τεχνολογία δημιουργούνται όλο και περισσότερο βελτιωμένα οπτικά συστήματα που καταγράφουν την κίνηση, την οποία αποτυπώνουν σε όλο και περισσότερο βελτιωμένους ψηφιακούς χαρακτήρες.

Το 1983, οι Ginsberg και Maxwell στο MIT παρουσίασαν τη γραφική μαριονέτα (graphical marionette), ένα οπτικό σύστημα καταγραφής της κίνησης, όπου μπορεί να καταγραφεί το animation μέσω της αναπαράστασης της κίνησης. Κατασκεύασαν, λοιπόν, ένα κοστούμι με leds σε διάφορα σημεία τους σώματος (αρθρώσεις κλπ.) και δύο κάμερες που κατέγραφαν εικόνες, κι έπειτα επέστρεφαν την δυσδιάστατη θέση καθενός led στο οπτικό τους πεδίο. Ο υπολογιστής δεχόταν τις πληροφορίες της θέσης από τις δύο κάμερες, για να αποκτήσει μια 3d παγκόσμια συντεταγμένη για κάθε led. Έτσι αφού υπήρχαν οι συντεταγμένες κάθε led που υπήρχε στο κοστούμι, το μόνο που έμενε ήταν η απεικόνισή του σε ένα ψηφιακό χαρακτήρα με βάσει τις αλλαγές στις συντεταγμένες των led.

Πάνω σε αυτή τη φιλοσοφία στηρίχτηκαν όλα τα μετέπειτα ανεπτυγμένα οπτικά συστήματα καταγραφής της κίνησης. Το 1985 ο Robert Abel υλοποίησε μια ρεαλιστικότερη κίνηση ενός ανθρώπου στο ρομπότ Brilliance της National Canned Food Information Council, χρησιμοποιώντας περισσότερες κάμερες και 18 μαύρες κουκίδες στο σώμα μιας γυναίκας ηθοποιού. Το 1988, η Pacific Data Images κατασκεύασε ένα ελαφρύ πλαστικό “εξωσκελετό” (που το ονόμασαν exoskeleton) για το μισό πάνω μέρος του σώματος ώστε να παρακολουθεί την κίνηση του άνω κορμού, του κεφαλιού και των χεριών, ώστε να μπορούν οι ηθοποιοί να ελέγχουν την κίνηση ψηφιακών χαρακτήρων.

Τις τελευταίες δεκαετίες, οι προσεγγίσεις για την ρεαλιστικότερη καταγραφή της ανθρώπινης κίνησης έχουν βελτιωθεί και εκσυγχρονιστεί, λόγω της ραγδαίας ανάπτυξης της τεχνολογίας. Η σημαντικότερη εξέλιξη, είναι η εισαγωγή νέων οπτικών μέσων, τα οποία μπορούν να καταγράψουν την κίνηση χωρίς τοποθέτηση δεικτών, led ή πολλαπλών καμερών. Μια απλή αλληλουχία εικόνων είναι αρκετή για να αποκομιστούν τα δεδομένα κίνησης.

Στο πλαίσιο της εισαγωγής νέων τεχνολογιών στον χώρο της καταγραφής της κίνησης, έρχονται και οι αισθητήρες κίνησης (αδρανειακοί αισθητήρες), οι οποίοι εισάγουν μια διαφορετική μέθοδο στην καταγραφή της κίνησης με χαμηλότερο κόστος. Αυτοί οι αισθητήρες αποτελούνται από ένα συνδυασμό ενός γυροσκοπίου, ενός μαγνητόμετρου και ενός επιταχυνσιόμετρου, ώστε να υπολογίζονται τα ποσοστά περιστροφής. Με την χρήση των αδρανειακών αισθητήρων δεν χρειάζονται πρόσθετες κάμερες, πομποί ή δείκτες για να περιγράψουν τη σχετική κίνηση, και οι πιο πολλοί από αυτούς τους αισθητήρες έχουν χαμηλό κόστος.

Προφανώς γυροσκόπια και αδρανειακοί αισθητήρες σε πρώιμο στάδιο υπήρχαν και παλαιότερα. Όμως η εισαγωγή των μικρο-ηλεκτρομηχανικών συστημάτων (MEMS), στα μέσα της δεκαετίας του '90, τα οποία χρησιμοποιούνταν κυρίως για τον έλεγχο των αερόσακων στα αυτοκίνητα έως τότε, έδωσαν νέες δυνατότητες στην καταγραφή της κίνησης μέσω φθηνών πια chip. Η βελτίωσή τους και η χρήση τους σε όλο και περισσότερες εφαρμογές, τα κατέστησε μια αρκετά οικονομική λύση για όσους ήθελαν να καταγράψουν την περιστροφή και την ακριβή θέση ενός σώματος στον τρισδιάστατο χώρο.

Πλέον χρησιμοποιούνται και οι 2 μέθοδοι ανίχνευσης της κίνησης (οπτικά συστήματα, αδρανειακοί αισθητήρες) σε υψηλό επίπεδο τεχνολογίας. Υπάρχουν δύο ακόμα όχι και τόσο διαδεδομένοι μέθοδοι στην καταγραφή της κίνησης, η μαγνητική, η οποία γίνεται με τη χρήση τριών ορθογώνιων πηνίων, καθώς και τα μηχανικά συστήματα καταγραφής, τα οποία αποτελούνται από μια σκελετοειδή δομή στο σώμα του ανθρώπου. Λόγω του χαμηλότερου κόστους στην παρούσα εργασία θα ασχοληθούμε με την καταγραφή της κίνησης μέσω αδρανειακών αισθητήρων.

1.3 Ανασκόπηση πτυχιακής εργασίας

Για τη δημιουργία της πτυχιακής εργασίας αρχικά έκανα έρευνα για να βρω ποια υλικά θα χρησιμοποιήσω. Έψαξα σχετικά χαμηλού κόστους υλικά, τα οποία όμως θα τηρούσαν τις προδιαγραφές της αρχικής ιδέας.

Εφόσον θα χρησιμοποιούσα μία Arduino πλακέτα ανοιχτού κώδικα και ήθελα να είναι σχετικά μικρού μεγέθους, αρχικά στράφηκα σε Arduino pro mini. Βλέποντας όμως ότι η συγκεκριμένη πλακέτα χρειάζεται FTDI καλώδιο για προγραμματισμό αντί για usb και θα χρειαζόμουν επιπλέον υλικό για φόρτιση της Li-Po μπαταρίας με usb, αποφάσισα να επιλέξω μια κεντρική πλακέτα η οποία θα έχει τη δυνατότητα να προγραμματιστεί με usb και με την ίδια υποδοχή θα φορτίζεται και η μπαταρία του κυκλώματος. Έτσι επέλεξα την Pro Trinket της Adafruit, η οποία είχε τη δυνατότητα μέσω μίας πλακέτας που συνδυάζεται με την pro trinket (Adafruit Pro trinket liion/lipoly backpack add-on) να συνδέεται κατευθείαν η μπαταρία και να φορτίζεται μέσω του usb. Επιπλέον, έψαξα ένα καλό αισθητήρα ανίχνευσης κίνησης, ο οποίος να δίνει 3 μετρήσεις γυροσκοπίου (3 axis gyroscope) και 3 μετρήσεις επιταχυνσιομέτρου (3-axis acceleration). Η επιλογή του MPU6050, τον οποίο και χρησιμοποίησα, έγινε για δύο λόγους. Πρώτον διότι ήταν αρκετά οικονομικός συγκριτικά με άλλους και δεύτερον γιατί διαθέτει έναν ενσωματωμένο επεξεργαστή ψηφιακής κίνησης (on board Digital Motion Processor), ο οποίος δίνει πιο ακριβείς και φιλτραρισμένες μετρήσεις, ενώ οι μετρήσεις που έρχονται κατευθείαν από τον αισθητήρα δεν είναι τόσο ακριβείς, καθώς έχουν θόρυβο και χρειάζονται την υλοποίηση ειδικών φίλτρων (Kalman ή complementary). Τέλος, όσον αφορά το κύκλωμα, έψαξα μια πλακέτα bluetooth για ασύρματη επικοινωνία και εν τέλει κατέληξα στην πιο διαδεδομένη στο πεδίο του Arduino, το HC-05.

Έπειτα ήθελα κάπως να θωρακίσω το κύκλωμα, το οποίο δημιούργησα. Έτσι βρήκα δυο μικρά πλαστικά κουτάκια ειδικά για κυκλώματα. Τοποθέτησα τα δύο κυκλώματά μου μέσα σε αυτά και άνοιξα δύο τρύπες στο καθένα, μία στο usb της κεντρικής πλακέτας ώστε να μπορεί να συνδεθεί με το κύκλωμα usb καλώδιο, είτε για φόρτιση της μπαταρίας είτε για προγραμματισμό της πλακέτας και μία για να περνάει ένας διακόπτης on/off, ο οποίος διακόπτει και επαναφέρει το ρεύμα από τη μπαταρία στο κύκλωμα και χρησιμοποιείται για να ανοίγουμε και να κλείνουμε το κύκλωμα.

Στη συνέχεια βρήκα τον κώδικα MPU6050_DMP6, ο οποίος υπήρχε στο διαδίκτυο για λήψη των DMP μετρήσεων του αισθητήρα και τον τροποποίησα ώστε να προσαρμοστεί σε ασύρματη επικοινωνία. Ύστερα βρίσκοντας το κατάλληλο 3d μοντέλο χεριού από το asset store του λογισμικού Unity και χρησιμοποιώντας το εν λόγω λογισμικό κατάφερα να πραγματοποιήσω την αναπαράσταση της κίνησης. Ένα script γραμμένο σε C# ανοίγει την ασύρματη σειριακή πόρτα που επικοινωνεί με το πρώτο κύκλωμα και ένα άλλο script ανοίγει την άλλη σειριακή πόρτα που επικοινωνεί με το δεύτερο κύκλωμα. Οι τιμές που λαμβάνουν τα script μεταβάλλουν τις x,y,z τιμές περιστροφής (rotation) στις δύο αρθρώσεις αντίστοιχα και έτσι το 3d μοντέλο περιστρέφεται σε πραγματικό χρόνο, προσδίδοντας πιστά την αναπαράσταση της κίνησης μας.

ΚΕΦΑΛΑΙΟ 2

ΤΕΧΝΟΛΟΓΙΑ ARDUINO

Σε αυτό το κεφάλαιο γίνεται μία εισαγωγή στον κόσμο του Arduino. Είναι σημαντικό να κατανοηθεί η τεχνολογία αυτή, καθώς αποτέλεσε σημαντικό κομμάτι της εργασίας που υλοποίησα.

2.1 Εισαγωγή στο Arduino

Η ιδέα του Arduino ξεκίνησε το 2005 όταν οι Massimo Banzi και David Cueartielles προσπάθησαν να φτιάξουν μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Έδωσαν το όνομα στο σχέδιο από τον Arduino της Ingea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας, την ίδια περιοχή στην οποία στεγαζόταν η εταιρεία υπολογιστών Olivetti.

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++) και διαθέτει ένα ολοκληρωμένο περιβάλλον ανάπτυξης. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

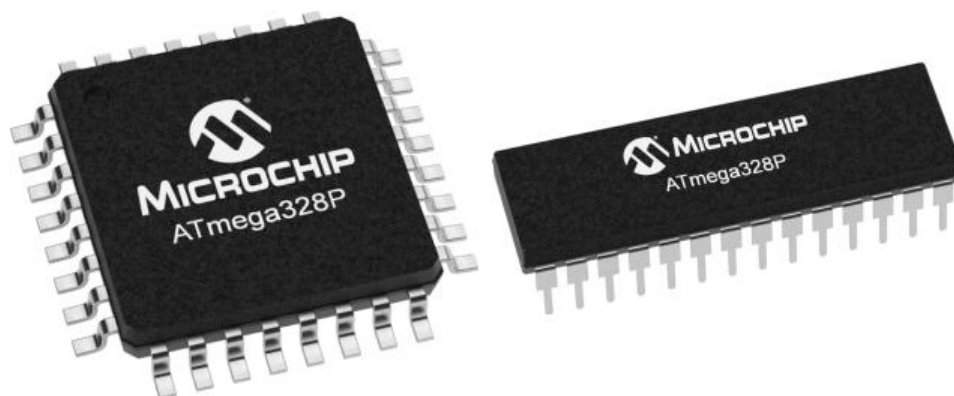
Το λογισμικό επίσης διατίθεται ελεύθερο, καθώς και χρήσιμα κομμάτια κώδικα προγραμματισμού τις πλακέτας για την υλοποίηση διάφορων εφαρμογών.



Εικόνα 2.1: Λογότυπο Arduino

2.2 Μικροελεγκτής Arduino

Οι Arduino πλακέτες χρησιμοποιούν μικροελεγκτές της Atmel AVR (ATmega328, ATmega168 κλπ.). Ο μικροελεγκτής ATmega328 ο οποίος είναι ο πιο κλασικός στη σειρά των Arduino, είναι ένας μικροεπεξεργαστής CMOS 8-bit βασισμένος στην αρχιτεκτονική RISC (reduced instruction set computer). Με την εκτέλεση των δυναμικών εντολών σε έναν μοναδικό κύκλο μηχανής, ο ATmega328 έχει αποδόσεις που πλησιάζουν το 1 MIPS (Million Instructions Per Second) ανά MHz που επιτρέπει στον σχεδιαστή να βελτιστοποιήσει την κατανάλωση ισχύος έναντι της ταχύτητας επεξεργασίας. Ακολουθεί εικόνα και πίνακας με τα χαρακτηριστικά του εν λόγω μικροελεγκτή:



Εικόνα 2.2: ATmega328

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	20
RAM Bytes	2,048
Data EEPROM (bytes)	1024
Digital Communication Peripherals	1-UART, 2-SPI, 1-I2C
Capture/Compare/PWM Peripherals	1 Input Capture, 1 CCP, 6PWM
Timers	2 x 8-bit, 1 x 16-bit
Comparators	1
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	32
Low Power	Yes

Πίνακας 2.1: Χαρακτηριστικά ATmega328

2.3 Εισαγωγή στα βασικά μοντέλα Arduino

Υπάρχουν αρκετές Arduino πλακέτες, οι οποίες έχουν διαφορετικά χαρακτηριστικά και δυνατότητες ώστε να ανταποκρίνονται σε μια πληθώρα διαφορετικών αναγκών. Παρακάτω ακολουθούν οι πιο βασικές, με εξαίρεση την πλακέτα την οποία χρησιμοποίησα στην συγκεκριμένη πτυχιακή, για την οποία θα υπάρξει εκτενής αναφορά στο επόμενο κεφάλαιο.

- *Arduino Uno*

Το Arduino Uno είναι η βασικότερη πλακέτα Arduino. Διαθέτει τον μικροελεγκτή ATmega328P και τον ATmega16U2 σαν μετατροπέα USB-to-serial, για σύνδεση με υπολογιστή. Μπορεί να λειτουργήσει με την τροφοδοσία του USB για ρεύμα έως 500 mA. Ενσωματώνει ένα LED το οποίο είναι συνδεδεμένο με το pin 13. Διαθέτει την είσοδο AREF μέσω της οποίας μπορεί να δοθεί εξωτερική τάση αναφοράς στον ενσωματωμένο A/D. Διαθέτει ένα πλήκτρο RESET. Διαθέτει δύο LED (RX και TX) τα οποία αναβοσβήνουν όταν υπάρχει επικοινωνία με τον υπολογιστή μέσω USB.



Εικόνα 2.3: Arduino Uno R3

- *Arduino Leonardo*

Το Arduino Leonardo χρησιμοποιεί τον μικροελεγκτή ATmega32u4. Έχει 20 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες 7 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 12 ως αναλογικές εισοδοί. Επίσης διαθέτει ένα ταλαντωτή κρυστάλλου 16 MHz, μία σύνδεση micro usb, μία υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset.



Εικόνα 2.4: Arduino Leonardo

- *Arduino Mega 2560*

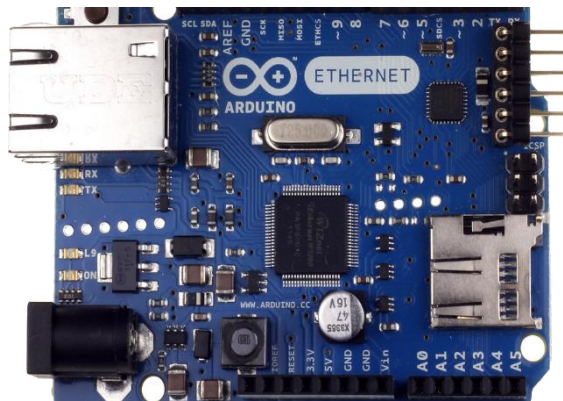
Το Arduino Mega χρησιμοποιεί τον μικροελεγκτή ATmega2560 και τον ATmega8U2 σαν μετατροπέα USB-to-serial. Έχει ολική μνήμη 256kB. Παραχωρεί 54 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες 14 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 16 ως αναλογικές εισοδοι. Επίσης διαθέτει ένα ταλαντωτή κρυστάλλου 16 MHz, 4 UARTS, μία σύνδεση micro usb, μία υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset.



Εικόνα 2.5: Arduino Mega

- *Arduino Ethernet*

Το Arduino Ethernet χρησιμοποιεί τον μικροελεγκτή που χρησιμοποιεί και το Arduino Uno τον ATmega328P. Έχει 14 ψηφιακές ακίδες εισόδου/εξόδου και 6 αναλογικές εισόδους. Επίσης διαθέτει ένα ταλαντωτή κρυστάλλου 16 MHz, μία υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset. Η διαφορά του Arduino Ethernet με τις άλλες πλακέτες είναι ότι δεν διαθέτει σύνδεση με usb, ενώ έχει microSD card reader.



Εικόνα 2.6: Arduino Ethernet

- *Arduino Yun*

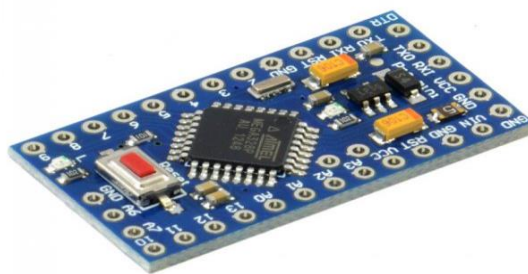
Το Arduino Yun συνδυάζει τη δύναμη του Linux με την ευκολία του Arduino. Χρησιμοποιεί τον μικροελεγκτή ATmega32u4 και το Atheros AR9331. Είναι συνδυασμός ενός κλασικού Arduino Leonardo βασισμένο στον ATmega32u4 με σύστημα Wi-Fi σε ένα chip που τρέχει Linino, μια διανομή Linux σε OpenWrt. Έχει 20 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες 7 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 12 ως αναλογικές εισοδοί. Επιπλέον, διαθέτει ένα ταλαντωτή κρυστάλλου 16 MHz, μία σύνδεση micro usb, μία υποδοχή τροφοδοσίας, μια ICSP και τρία κουμπιά reset. Όπως αναφέραμε διαθέτει Wi-Fi, όπως επίσης και Ethernet.



Εικόνα 2.7: Arduino Yun

- *Arduino Mini*

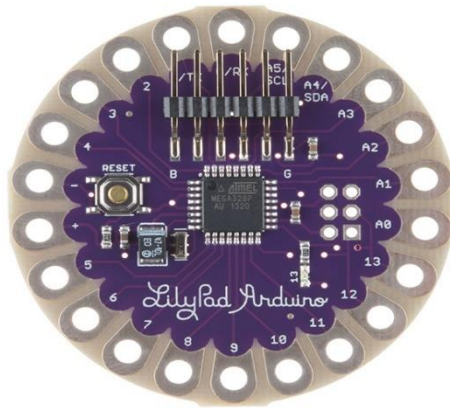
Το Arduino Mini χρησιμοποιεί τον μικροελεγκτή ATmega328P, ενώ αρχικά χρησιμοποιούσε τον ATmega168. Έχει 14 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως έξοδο PWM και 8 ως αναλογικές εισοδοί. Επίσης διαθέτει ένα ταλαντωτή κρυστάλλου 16 MHz και ένα κουμπί reset. Προγραμματίζεται με FTDI to usb adapter ή με τη βοήθεια του Arduino usb 2 serial micro.



Εικόνα 2.8: Arduino Mini

- *Arduino Lilypad*

Το Arduino Lilypad χρησιμοποιεί τον μικροελεγκτή ATmega328P ή ATmega168. Χρησιμοποιείται για εφαρμογές ένδυσης καθώς είναι μινιμαλιστικό και μικροσκοπικό. Έχει 14 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως έξοδο PWM και 12 ως αναλογικές εισοδοί. Η τάση λειτουργίας της πλακέτας είναι από 2V έως 5V.



Εικόνα 2.9: Arduino Lilypad

2.4 Λογισμικό Arduino

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ.

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++ και ονομάζονται sketches. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Το IDE του Arduino χρησιμοποιεί

Ασύρματη καταγραφή κίνησης χεριού

το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

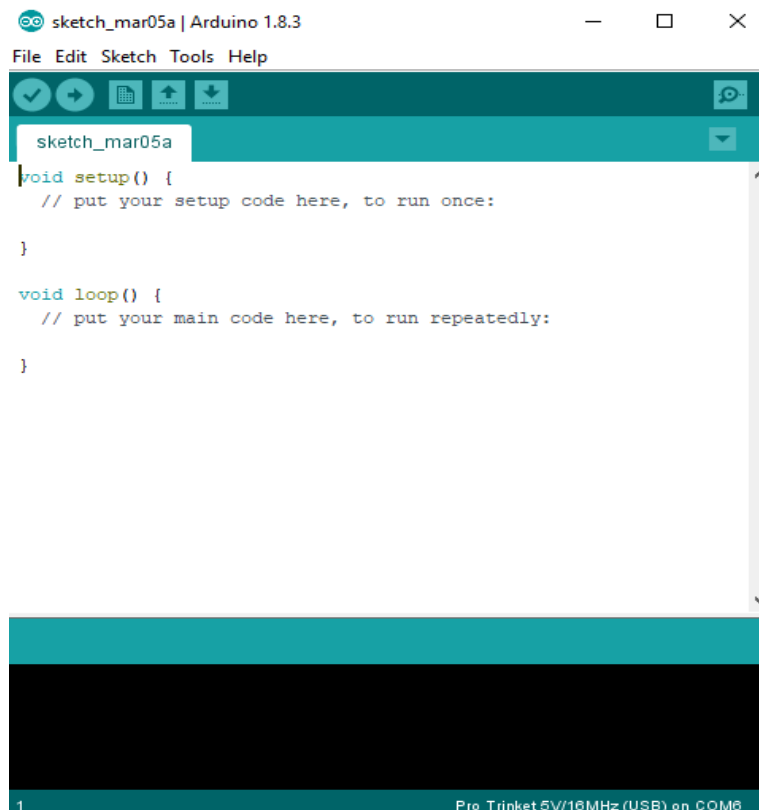
Οι χρήστες πρέπει μόνο να ορίσουν δύο βασικές συναρτήσεις για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup():μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος, η οποία αρχικοποιεί τις ρυθμίσεις

-loop():μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Πάνω από αυτές τις συναρτήσεις δηλώνονται οι βιβλιοθήκες που θα χρησιμοποιηθούν από το πρόγραμμά μας.

Δεδομένου ότι η πλατφόρμα Arduino είναι ανοιχτού κώδικα, το λογισμικό της διατίθεται δωρεάν στη διεύθυνση: <https://www.arduino.cc/en/main/software>



```
sketch_mar05a | Arduino 1.8.3
File Edit Sketch Tools Help
sketch_mar05a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
1 Pro Trinket 5V/16MHz (USB) on COM8
```

Εικόνα 2.10: Arduino sketch

2.5 Πλεονεκτήματα Arduino

Το Arduino δεν αποτελεί ένα απλό μικροελεγκτή. Αποτελεί μία ιδέα που έχει ως φιλοσοφία την εύκολη επικοινωνία μεταξύ υλικού (hardware) και λογισμικού (software). Έτσι, διαφέρει από τους υπόλοιπους μικροελεγκτές, καθώς απλοποιεί τη διαδικασία χρήσης τους και προγραμματισμού τους. Είναι, λοιπόν, κατάλληλο για αρχάριους, αφού προσφέρει αναλυτικές οδηγίες χρήσης πλακετών και λογισμικό στο διαδίκτυο, καθώς και για διδακτική ύλη καθηγητών σε μία πρώτη γνωριμία με τους μικροελεγκτές.

Όσων αφορά το κομμάτι του υλικού (hardware), πρόκειται για μία πολύ συμφέρουσα λύση καθώς προσφέρει μία πληθώρα διαφορετικών πλακετών, οι οποίες ανταποκρίνονται σε μία πληθώρα διαφορετικών αναγκών. Επίσης προσφέρει μία τεράστια γκάμα από αισθητήρες που ανταποκρίνονται σε πολλών ειδών project και οι οποίοι είναι σχεδιασμένοι με τέτοιο τρόπο, ώστε να έχουν εύκολα μία άριστη επικοινωνία με τις κεντρικές πλακέτες. Επιπλέον, τα Arduino διαθέτουν μία τεχνολογία εξαρτημάτων, τα οποία είναι σχεδιασμένα ώστε να συνδέονται απευθείας με όλα τα pin του Arduino, τα λεγόμενα shields, προεκτείνοντας έτσι τη λειτουργικότητά τους. Τέλος, αποτελεί μια value for money επιλογή, καθώς έχει χαμηλότερο κόστος σε σχέση με άλλους μικροελεγκτές.

Στο κομμάτι του λογισμικού (software), όπως προαναφέρθηκε, προσφέρει επίσης πολλά πλεονεκτήματα. Τα Arduino χρησιμοποιούν μικροελεγκτές της Atmel AVR, οι οποίοι μπορούν πολύ εύκολα να προγραμματιστούν μέσω του λογισμικού ανάπτυξης του Arduino (Arduino IDE), το οποίο είναι ένα περιβάλλον, πολύ εύχρηστο για αρχάριους χρήστες και αρκετά ευέλικτο για πιο προχωρημένους. Όπως αναφέραμε, διατίθεται δωρεάν κι υπάρχουν πάρα πολλές οδηγίες και βιβλιοθήκες στο διαδίκτυο, ώστε να μπορέσουμε να το χρησιμοποιήσουμε εύκολα και γρήγορα. Τέλος, έχει έναν compiler για την μεταγλώττιση των sketch, συντακτική χρωματική σήμανση, εμφάνιση σφαλμάτων κατά τη μεταγλώττιση και μία σειριακή οθόνη (serial monitor) για παρακολούθηση των αποτελεσμάτων της σειριακής επικοινωνίας της πλακέτας μας με τον υπολογιστή.

ΚΕΦΑΛΑΙΟ 3

ΚΑΤΑΣΚΕΥΗ

Σε αυτό το κεφάλαιο αναλύεται ο τρόπος λειτουργίας των υλικών που χρησιμοποιήθηκαν, καθώς και ο τρόπος σύνδεσής τους για την τελική μορφή της κατασκευής.

3.1 Τεχνολογίες που χρησιμοποιήθηκαν

Η αρχική ιδέα της κατασκευής ήταν ένα Arduino κύκλωμα με μία κεντρική Arduino πλακέτα, η οποία να συνδέεται: με έναν αισθητήρα που θα καταγράφει την κίνηση και ένα bluetooth εξάρτημα, το οποίο θα μεταδίδει ασύρματα τις μετρήσεις αυτές. Λόγω του ότι η επικοινωνία έπρεπε να είναι ασύρματη κι όχι μέσω usb, η τροφοδοσία θα πρόεκυπτε από μια μικρή μπαταρία. Τέλος, ένας μικρός διακόπτης θα ήταν χρήσιμος για άνοιγμα και κλείσιμο του κυκλώματος (on/off).

Επομένως, η πρώτη επιλογή που έπρεπε να κάνω ήταν αυτή της κεντρικής πλακέτας Arduino. Ανάμεσα σε διάφορες πλακέτες Arduino (ορισμένες από τις οποίες προαναφέρθηκαν στο προηγούμενο κεφάλαιο) διαπίστωσα ότι η Adafruit έχει δημιουργήσει μία πλακέτα η οποία είναι πιο κοντά από κάθε άλλη στα θέλω μου, την Pro Trinket των 5V. Πρόκειται για μία Arduino πλακέτα, η οποία είναι πολύ μικρή σε μέγεθος (σημαντικό για την εργασία μου), διαθέτει τον κλασικό μικροελεγκτή ATmega328P, ο οποίος μπορεί εύκολα να προγραμματιστεί με mini usb. Τέλος, μέσω μίας μικρής πλακέτας (Adafruit Pro Trinket Liion/Lipoly Backpack Add-on), η οποία συμπληρώνει την πρώτη, δίνεται η δυνατότητα άμεσης σύνδεσης μπαταρίας για τροφοδοσία, φόρτισης μπαταρίας μέσω του usb της κεντρικής πλακέτας, καθώς και σύνδεσης διακόπτη on/off για το κύκλωμα. Επομένως τα έχουμε όλα σε ένα και σε σχετικά μικρό χώρο.

Δεύτερη επιλογή που είχα να κάνω, ήταν ο αδρανειακός αισθητήρας που θα χρησιμοποιούσα, ο οποίος θα ανίχνευε την κίνηση. Αρχικά, διαβάζοντας θεωρητικό υλικό διαπίστωσα ότι χρειαζόμουν έναν αισθητήρα που θα δίνει 3 μετρήσεις γυροσκοπίου (gyro-x, gyro-y, gyro-z) και 3 μετρήσεις επιταχυνσιόμετρου (accel-x, accel-y, accel-z). Ένας πολύ καλός και φθηνός αισθητήρας, ο οποίος δίνει αυτές τις

μετρήσεις και με μεγάλη ακρίβεια είναι ο MPU-6050. Επίσης διαθέτει έναν επεξεργαστή, ο οποίος μέσω ειδικών συναρτήσεων δίνει φιλτραρισμένες τιμές, οπότε δεν θα χρειαζόταν να φτιάξω κάποιο φίλτρο, κι έτσι επέλεξα αυτόν τον αισθητήρα.

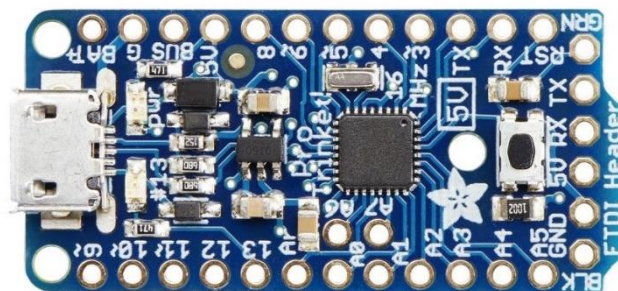
Επόμενο βήμα, ήταν η επιλογή ενός bluetooth module για να μεταδώσω ασύρματα τις τιμές στον υπολογιστή. Ο πιο κλασσικός και απλός σε συνδεσμολογία με Arduino ήταν ο HC-05, οπότε και τον επέλεξα.

Τέλος, για τροφοδοσία επέλεξα μία μπαταρία πολυμερών λιθίου (Li-po), η οποία είναι επαναφορτιζόμενη, έχει χαμηλό βάρος, πολύ μικρό μέγεθος και αντοχή στην υπερφόρτιση.

3.2 Μητρική πλακέτα κυκλώματος

Μητρική πλακέτα του κυκλώματός μου θεωρείται η κεντρική πλακέτα Arduino που χρησιμοποίησα καθώς είναι η πλακέτα που έχει τον μικροελεγκτή, τον οποίο προγραμμάτισα για να κάνει τις λειτουργίες τις οποίες ήθελα και όλες οι υπόλοιπες πλακέτες είναι συμπληρωματικές, καθώς συνδέθηκαν με αυτή. Η μητρική μου πλακέτα λοιπόν είναι η Adafruit Pro Trinket (5V, 16MHz).

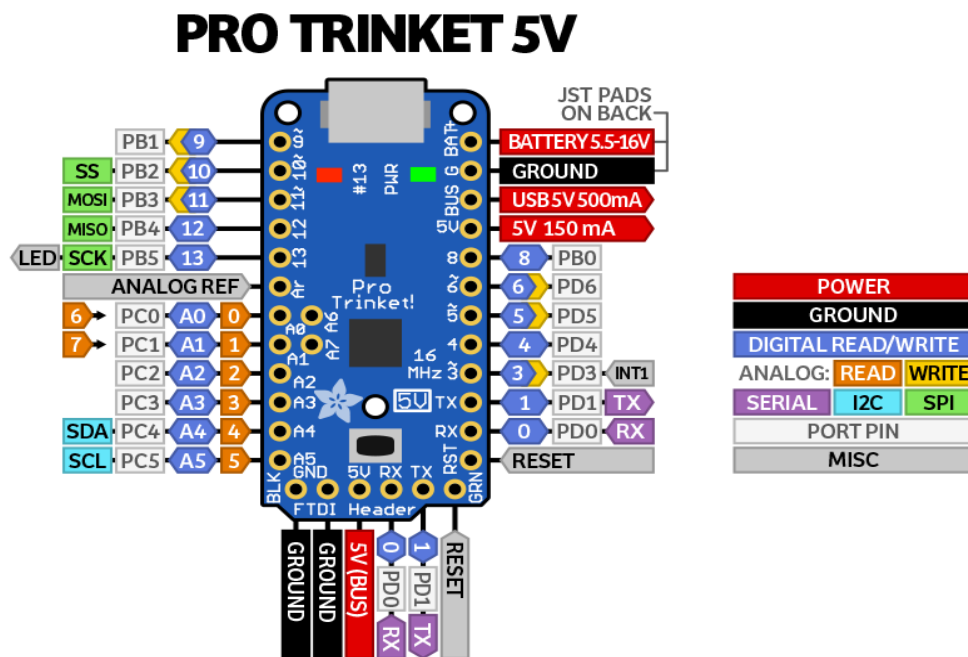
3.2.1 Adafruit Pro trinket



Εικόνα 3.1: Adafruit Pro Trinket 5V

Όπως προαναφέρθηκε, το Adafruit Pro Trinket διαθέτει τον μικροελεγκτή ATmega328P, με συχνότητα (clock rate) 16MHz και μνήμη φλας (flash memory)

28K. Επίσης έχει usb bootloader για προγραμματισμό του ATmega328P με το λογισμικό του Arduino (IDE) και ένα FTDI Header, δηλαδή μια σειρά ακίδων που δίνουν τη δυνατότητα επαναπρογραμματισμού με FTDI σύνδεση. Ο ρυθμιστής ρεύματος (power regulator) 5V μπορεί να παρέχει έξοδο (output) 150mA και εκτός από μπαταρία μπορούμε να έχουμε τροφοδοσία από σύνδεση με usb. Τέλος, διαθέτει ένα κουμπί επανεκκίνησης (reset) για είσοδο στον bootloader ή για επανεκκίνηση του προγράμματος.



Εικόνα 3.2: Διάγραμμα ακίδων (pinout) Adafruit Pro Trinket 5V

Όπως μπορούμε να παρατηρήσουμε από το διάγραμμα, οι ακίδες του Adafruit Pro Trinket μπορούν να χωριστούν σε τέσσερις κατηγορίες:

- τις ακίδες ενέργειας (power pins)
- τις ακίδες εισόδου εξόδου (ψηφιακές, αναλογικές και ψηφιακές/αναλογικές)
- τον FTDI Header
- τις υπόλοιπες ακίδες

Οι ακίδες ενέργειας είναι τέσσερις (BAT+, G, BUS, 5V). Η ακίδα BAT+ είναι το ρεύμα (θετικός πόλος) της μπαταρίας. Επομένως είναι το σημείο στο οποίο

συνδέουμε το θετικό πόλο της μπαταρίας ή του αντάπτορα που θα χρησιμοποιήσουμε. Η ακίδα G (ground), είναι η γείωση, δηλαδή το σημείο που συνδέεται όλες οι γειώσεις, είτε ο αρνητικός (-) πόλος της μπαταρίας, είτε η γείωση του κυκλώματος. Η ακίδα BUS (USB Bus +Output) είναι η έξοδος από το usb (5V). Όταν δηλαδή είναι συνδεδεμένο το usb διαρρέεται από ρεύμα και χρησιμοποιείται σαν έξοδος ρεύματος (πχ. για φόρτιση μπαταρίας) και όχι για είσοδος. Τέλος, η ακίδα 5V είναι έξοδος ρεύματος και διαρρέεται από ρεύμα είτε η πλακέτα τροφοδοτείται από usb, είτε από μπαταρία.

Οι ψηφιακές ακίδες εισόδου εξόδου (input/output pins) είναι 18 (RX, TX, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19), από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM (3, 5, 6, 9, 10, 11) και 6 ως αναλογικές έξοδοι (14, 15, 16, 17, 18, 19). Η ακίδα RX είναι η σειριακή είσοδος ενώ η TX η σειριακή έξοδος, γνωστές στην Arduino οικογένεια και ως Digital #0 και Digital #1 αντίστοιχα. Η ακίδα 3 είναι η interrupt #1, η αντίστοιχη με την 2 του Arduino Uno. Οι ακίδες 11, 12 και 13 είναι οι ακίδες για SPI (Serial Peripheral Interface) επικοινωνία MOSI, MISO και CLOCK αντίστοιχα. Ακίδες 2 και 7 δεν υπάρχουν γιατί χρησιμοποιούνται στον bootloader του usb. Οι ψηφιακές ακίδες 14-19 είναι οι αντίστοιχες A0-A5, οι 6 ακίδες δηλαδή που γίνονται και αναλογικές. Συγκεκριμένα η A4 και η A5 χρησιμοποιούνται για I2C (Inter-integrated Circuit) επικοινωνία, ως SDA (για data) και SCL (για παλμούς clock) αντίστοιχα. Οι αναλογικές ακίδες εισόδου εξόδου είναι 2 η A6 και η A7 και χρησιμοποιούνται μόνο ως αναλογικές.

Οι 6 κάτω ακροδέκτες αποτελούν το *FTDI Header* της πλακέτας (Future Technology Devices International). Έχει 2 ακίδες γείωσης (Ground), μία τροφοδοσίας (5V), 2 ακίδες για σειριακή επικοινωνία με είσοδο RX και έξοδο TX και μία ακίδα επανεκκίνησης (reset).

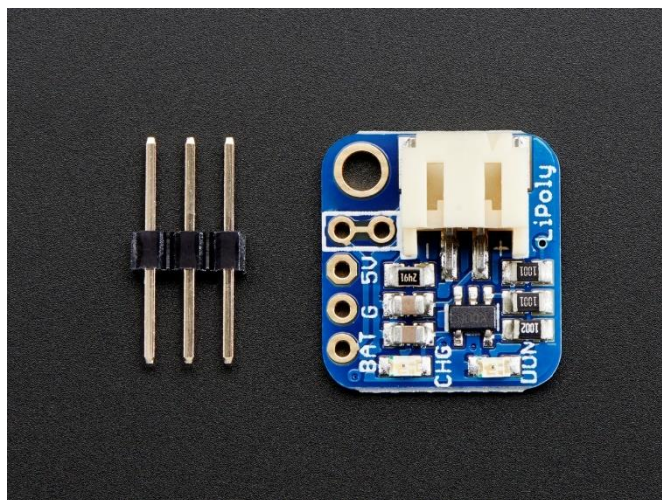
Οι υπόλοιπες ακίδες είναι μια ακόμα ακίδα επανεκκίνησης (reset) και η ακίδα Ar (analog reference), για τον αναλογικό μετατροπέα και τάση διαφορετική από 3V ή 5V.

3.2.2 Πλεονεκτήματα Adafruit Pro trinket

Η επιλογή της συγκεκριμένης πλακέτας έγινε διότι είναι μια πλακέτα που συνδυάζει πολλές δυνατότητες, οι οποίες ήταν απαραίτητες για την εργασία μου. Είναι μία πλακέτα, η οποία διαθέτει έναν κλασικό Arduino μικροελεγκτή, τον ATmega328P και προγραμματίζεται πολύ εύκολα με το λογισμικό του Arduino. Επιπλέον, πολύ σημαντικό είναι ότι αποτελεί μία πολύ μικρή πλακέτα σε μέγεθος και μπορεί να προγραμματιστεί μέσω usb συνδεσμολογίας, σε αντίθεση με την αντίστοιχη Arduino πλακέτα την Pro mini, η οποία θέλει ειδικό αντάπτορα (FTDI to usb) για να προγραμματιστεί. Έτσι εξοικονόμησα χώρο στο τελικό κύκλωμα, καθώς και κόστος (ναι μεν πιο ακριβή πλακέτα αλλά δεν χρειάζεται έξτρα κόστος για αντάπτορα). Τέλος, η συγκεκριμένη πλακέτα παρέχει μια φθηνή συμπληρωματική πλακέτα, την Adafruit Pro Trinket Liion/Lipoly Backpack Add-on, η οποία χρησιμοποιείται για τη φόρτιση της μπαταρίας του κυκλώματος κατευθείαν από το usb της κεντρικής πλακέτας και διαθέτει ακίδες για διακόπτη on/off του κυκλώματος. Επομένως, μέσω μίας υποδοχής μπορώ να φορτίζω την μπαταρία του κυκλώματός μου καθώς και να ανεβάζω κώδικα.

3.2.3 Adafruit Pro trinket liion/lipoly backpack add-on

Όπως προανέφερα, η πλακέτα Adafruit Pro Trinket Liion/Lipoly Backpack Add-on αποτελεί μία μικρή συμπληρωματική πλακέτα της Adafruit Pro Trinket.



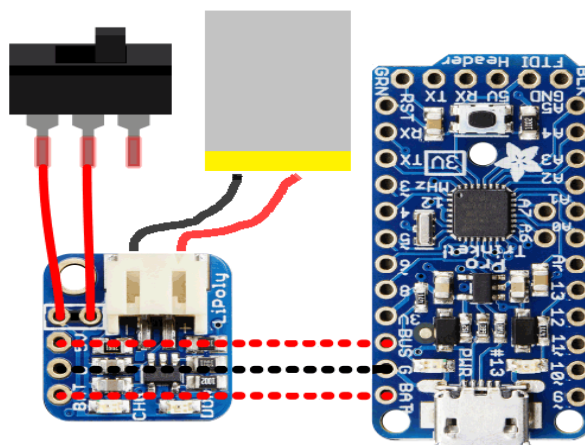
Εικόνα 3.3: Adafruit Pro Trinket Liion/Lipoly Backpack Add-on

Όπως μπορούμε να παρατηρήσουμε από την εικόνα, η συγκεκριμένη πλακέτα διαθέτει 3 ακροδέκτες, οι οποίοι τοποθετούνται στις 3 της ακίδες με την ίδια σειρά όπως και στην κεντρική πλακέτα Adafruit Pro Trinket (BAT, G, 5V). Αυτό συμβαίνει διότι η συγκεκριμένη πλακέτα προορίζεται για να τοποθετηθεί ακριβώς πάνω από την κεντρική πλακέτα και οι 3 ακίδες (BAT, G, 5V) της μίας να συνδέονται με τις αντίστοιχες της άλλης.

Έτσι έχουμε την ακίδα 5V, η οποία διαρρέεται από ρεύμα όταν η κεντρική πλακέτα είναι συνδεδεμένη με usb κι έτσι φορτίζεται η μπαταρία. Έπειτα έχουμε την ακίδα G, η οποία αποτελεί τη γείωση και συνδέεται με τη γείωση της κεντρικής πλακέτας καθώς και στις μπαταρίας. Τέλος, έχουμε την ακίδα BAT, η οποία είναι η έξοδος από τη μπαταρία μας και διαρρέεται από ρεύμα όσο ο διακόπτης είναι ανοιχτός και η μπαταρία είναι φορτισμένη.

Οι 2 ακίδες οι οποίες είναι κάθετες στην τριάδα (BAT, G, 5V) είναι το σημείο στο οποίο παρεμβάλλεται ο διακόπτης on/off ο οποίος ανοιγοκλείνει το κύκλωμά μας. Σε αυτές λοιπόν τις ακίδες, αφού αφαιρέσουμε με κάτι αιχμηρό μια μεταλλική μεμβράνη που συνδέει τις 2 ακίδες (θέλουμε να μη συνδέονται μεταξύ τους), συνδέουμε τυχαία την μία και την άλλη άκρη του διακόπτη με κάθε μία από αυτές αντίστοιχα, για να δίνουμε ή να κόβουμε το ρεύμα που δίνει η μπαταρία στο κύκλωμα.

Μία υποδοχή με 2 ακροδέκτες (+, -) είναι το σημείο στο οποίο τοποθετούνται οι 2 πόλοι της μπαταρίας μας.



Εικόνα 3.4: Συνδεσμολογία Adafruit Pro Trinket Liion/Lipo Backpack Add-on

3.3 Καταγραφή Κίνησης

Σε αυτό το υποκεφάλαιο θα μελετήσουμε τον τρόπο καταγραφής της κίνησης, με τον οποίο ασχοληθήκαμε στη συγκεκριμένη πτυχιακή εργασία, δηλαδή την καταγραφή κίνησης με αδρανειακούς αισθητήρες. Θα επισημάνουμε ορισμένες θεωρητικές έννοιες γύρω από τα αδρανειακά συστήματα αναφοράς, τον τρόπο λειτουργίας αδρανειακών αισθητήρων, καθώς και τον τρόπο λειτουργίας του αισθητήρα που χρησιμοποιήθηκε.

3.3.1 Εισαγωγή στην καταγραφή της κίνησης

Η καταγραφή κίνησης (motion capture ή mo-cap) είναι η διαδικασία της εγγραφής της κίνησης ενός αντικειμένου ή ανθρώπου. Χρησιμοποιείται παντού στο σύγχρονο κόσμο, από τον αθλητισμό και την ιατρική, μέχρι τον στρατό και τη ρομποτική. Επίσης χρησιμοποιείται για την επαλήθευση της μηχανικής όρασης, καθώς και για ψυχαγωγία. Πολύ σημαντική είναι η συμβολή της καταγραφής κίνησης στη διαδικασία ανάπτυξης ταινιών και βιντεοπαιχνιδιών, στην οποία πραγματοποιείται εγγραφή των δράσεων των ανθρώπων ηθοποιών και έπειτα αυτή η πληροφορία χρησιμοποιείται για την κίνηση των μοντέλων των ψηφιακών χαρακτήρων (3d model) σε τρισδιάστατο ή δισδιάστατο animation.

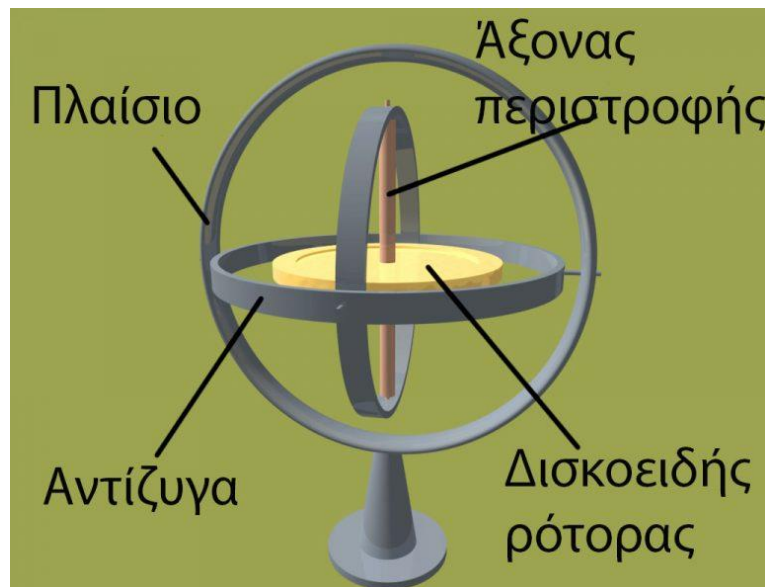
Όταν περιλαμβάνονται το πρόσωπο και τα δάχτυλα ή ήπιες εκφράσεις, συχνά ονομάζεται σύλληψη ερμηνείας (performance capture). Σε πολλούς τομείς, η σύλληψη κίνησης συχνά ονομάζεται παρακολούθηση κίνησης (motion tracking), αλλά στην κινηματογράφηση και τα βιντεοπαιχνίδια, η παρακολούθηση κίνησης συνήθως αναφέρεται περισσότερο στην τεχνική κινηματογράφησης, match moving.

Επομένως, η καταγραφή της κίνησης εν γένη μπορεί να πραγματοποιηθεί είτε με οπτικά μέσα (optical motion capture systems) όπως κάμερες ανίχνευσης κίνησης, είτε με αδρανειακούς αισθητήρες (IMUs). Οι αισθητήρες αυτοί, με τους οποίους θα ασχοληθούμε στην παρούσα πτυχιακή εργασία, είναι τα κύρια συστατικά ενός αδρανειακού συστήματος, τα οποία χρησιμοποιούν μετρήσεις γυροσκοπίων, επιταχυνσιόμετρων και μαγνητόμετρων κι έτσι ανιχνεύονται μεταβολές στην περιστροφική κίνηση και εκτιμάται η ακριβής θέση του σώματος στον τρισδιάστατο χώρο.

3.3.2 Γυροσκόπιο (gyroscope)

Γυροσκόπιο είναι μία συσκευή με μία περιστρεφόμενη μάζα, η οποία τείνει να διατηρεί σταθερό στο χώρο το επίπεδο περιστροφής της, εξαιτίας της αρχής της στροφορμής. Εφευρέθηκε από τον Ζαν Μπερνάρ Λεόν Φουκώ το 1852, ο οποίος προσπάθησε με αυτή τη συσκευή να αποδείξει την περιστροφή της Γης.

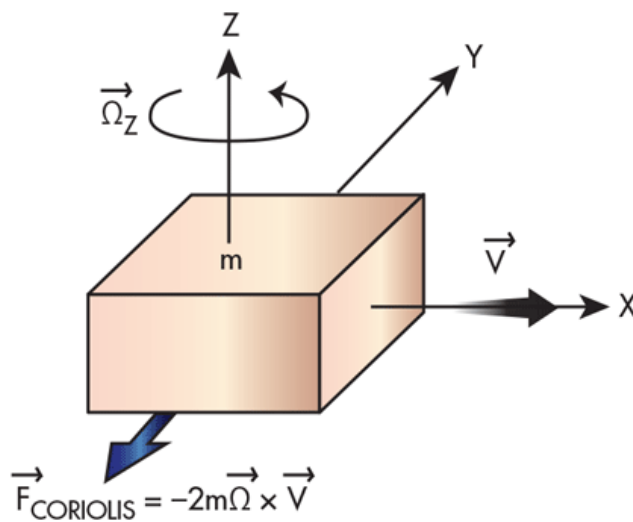
Το γυροσκόπιο αποτελείται από ένα δισκοειδή ρότορα αναρτημένο σε ελαφρούς δακτυλίους, τα αντίζυγα, που συνδέονται με το πλαίσιο υποστήριξης μέσω τριβένων (ρουλεμάν) σχεδόν μηδενικής τριβής. Περιστρεφόμενο σε υψηλές ταχύτητες το γυροσκόπιο ενεργεί με εξαιρετική σταθερότητα κρατώντας την κατεύθυνση του άξονα περιστροφής του ρότορα.



Εικόνα 3.5: Γυροσκόπιο

Το γυροσκόπιο είναι λοιπόν μια συσκευή, η οποία μετράει ταχύτητα περιστροφής, χρησιμοποιώντας τις αρχές αδράνειας και διατήρησης στροφορμής. Έτσι χρησιμοποιώντας την έλξη της βαρύτητας βρίσκει τον προσανατολισμό ενός σώματος. Όταν το σώμα είναι ακίνητο οι τιμές του είναι μηδενικές. Είναι λοιπόν ιδανικό για τη μέτρηση γωνιακών περιστροφών, καθώς και για τη μέτρηση δεδομένων που αφορούν την ακριβή κατεύθυνση και τον ρυθμό περιστροφής ενός σώματος ως προς κάθε άξονα περιστροφής.

Εκτός από τα κλασσικά μηχανικά γυροσκόπια, έχουμε τα γυροσκόπια δακτυλίου laser (ring laser gyroscopes), γυροσκόπια οπτικής ίνας (fiber optic gyroscopes), κβαντικά γυροσκόπια (quantum gyroscopes), καθώς και αυτά με τα οποία θα ασχοληθούμε και στην παρούσα εργασία τα μικροηλεκτρομηχανικά γυροσκόπια (MEMS gyroscopes). Αυτά τα σύγχρονα γυροσκόπια 3 αξόνων (3-axis gyroscope) μας δίνουν μία καλή εικόνα για τη γωνιακή ταχύτητα γύρω από τους κάθετους άξονές τους. Τα MEMS γυροσκόπια στηρίζουν τη λειτουργία τους στο φαινόμενο Coriolis. Μια μάζα (m) κινείται με ταχύτητα (v) σε ένα πλαίσιο αναφοράς που περιστρέφεται με γωνιακή ταχύτητα (Ω). Αυτή η μάζα δέχεται τη δύναμη Coriolis (F_{coriolis}). Η μάζα ταλαντεύεται γύρω από έναν οδηγό άξονα κι όταν το γυροσκόπιο περιστραφεί τότε προκαλείται μια δεύτερη ταλάντωση στον κάθετο άξονα ανίχνευσης λόγω της δύναμης Coriolis. Η γωνιακή ταχύτητα μπορεί να υπολογιστεί από τη μέτρηση αυτής της ταλάντωσης.



Εικόνα 3.6: Δύναμη Coriolis

3.3.3 Επιταχυνσιόμετρο (accelerometer)

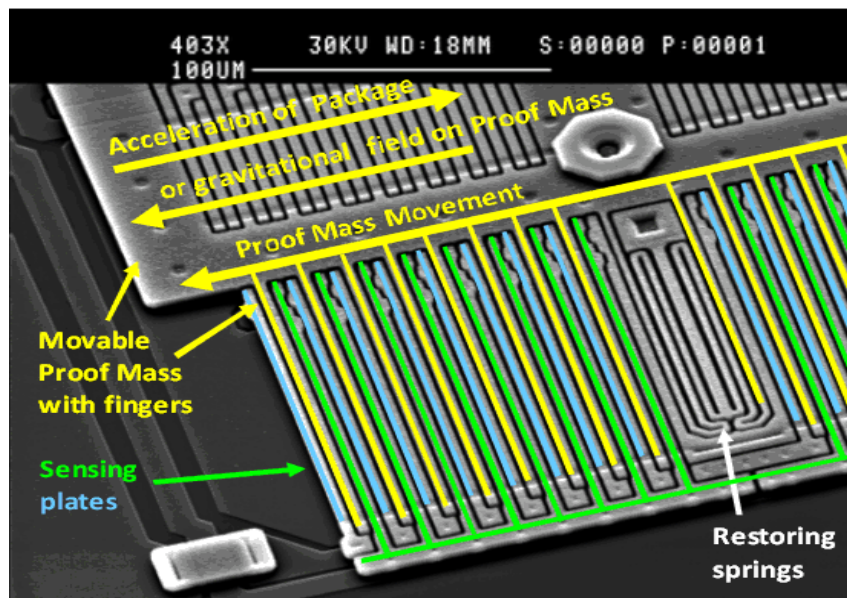
Το Επιταχυνσιόμετρο είναι μια συσκευή, η οποία έχει την ικανότητα να μετρά δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως είναι η επιτάχυνση της βαρύτητας, ή δυναμικές όταν προκαλούνται από αλλαγές στην

ταχύτητα ή στην διεύθυνση της κίνησης (επιταχύνσεις, επιβραδύνσεις, στροφές). Με αυτά λοιπόν, μπορούμε να μετρήσουμε τη γραμμική επιτάχυνση ενός σώματος, κάτι το οποίο μπορεί να μας βοηθήσει στον υπολογισμό της ταχύτητας και της θέσης ενός αντικειμένου στο χώρο. Κάτι το οποίο δεν μπορούμε να υπολογίσουμε με το επιταχυνσιόμετρο είναι περιστροφή (rotation). Την ταχύτητα περιστροφής στους 3 άξονες τη μετράμε με το γυροσκόπιο.

Οι μετρήσεις του επιταχυνσιόμετρου συμπεριλαμβάνουν και την στατική επιτάχυνση της βαρύτητας, το οποίο σημαίνει ότι ακόμα και όταν το σώμα είναι ακίνητο αυτό μας δίνει τιμές διάφορες του μηδενός, ενώ όταν το σώμα πραγματοποιεί ελεύθερη πτώση οι τιμές είναι μηδενικές.

Τα επιταχυνσιόμετρα τριών αξόνων (3-axis accelerometer) χρησιμοποιούνται στους αδρανειακούς αισθητήρες για τη μέτρηση της κλίσης ενός σώματος ως προς το οριζόντιο επίπεδο (επιφάνεια της γης).

Υπάρχουν πολλοί τρόποι με τους οποίους λειτουργούν τα επιταχυνσιόμετρα. Αυτός που μας ενδιαφέρει καθώς χρησιμοποιείται στους αδρανειακούς αισθητήρες είναι με τη χρήση κινούμενης μεταλλικής πλάκας και πλακών ανίχνευσης.

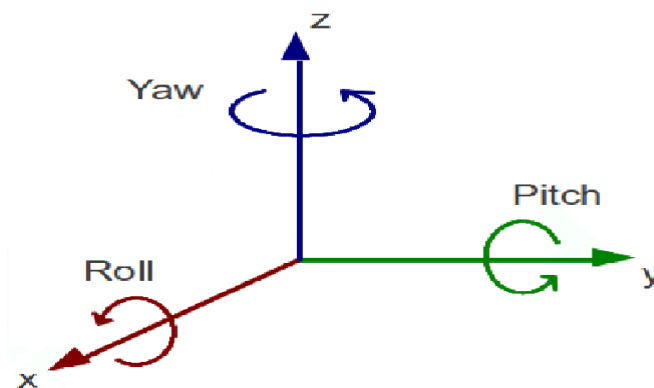


Εικόνα 3.7: Επιταχυνσιόμετρο

Μία μετακινούμενη μεταλλική μάζα βρίσκεται ανάμεσα σε μεταλλικές πλάκες ανίχνευσης. Μια επιτάχυνση της μάζας είτε του σώματος, είτε από το πεδίο βαρύτητας μετακινεί τη μεταλλική μάζα. Η απόκλιση ανάμεσα σε μεταλλική μάζα και πλάκα ανίχνευσης μετράται ως μεταβολή της χωρητικότητας. Το εσωτερικό κύκλωμα μετατρέπει αυτή τη μικροσκοπική χωρητικότητα σε ένα σήμα τάσης το οποίο ψηφιοποιείται και παράγεται ως έξοδος του επιταχυνσιομέτρου.

3.3.4 Γωνίες Euler

Οι γωνίες Euler είναι 3 γωνίες, οι οποίες εισήχθησαν στον χώρο των μαθηματικών από τον Leonhard Euler και χρησιμοποιούνται για να περιγράψουν τον προσανατολισμό ενός σώματος σε σχέση με ένα σταθερό σύστημα συντεταγμένων. Πιο συγκεκριμένα οι 3 αυτές γωνίες δίνουν ακριβή εκτίμηση της περιστροφής ενός αντικειμένου στους 3 άξονες του χώρου, επομένως μας βοηθούν στην εύρεση προσανατολισμού του σώματος. Στους 3 άξονες του συστήματος συντεταγμένων (x, y, z) έχουμε αντίστοιχα τις γωνίες κύλισης (roll), ανύψωσης (pitch), στροφής (yaw) με συμβολισμούς γωνιών (α , β , γ) ή (φ , θ , ψ).

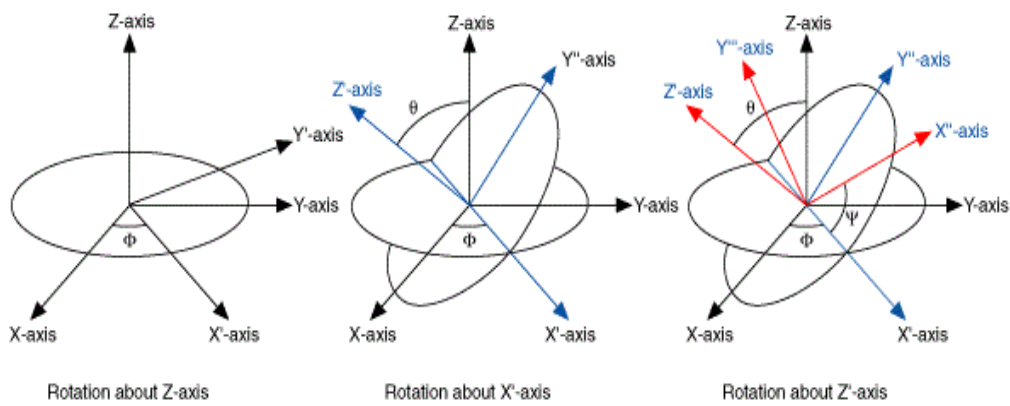


Εικόνα 3.8: Yaw-Pitch-Roll

Για να μελετήσουμε την περιστροφή ενός αντικειμένου στον τρισδιάστατο χώρο, πρέπει να ορίσουμε 3 συστήματα αναφοράς: το πλαίσιο αναφοράς του σώματος το οποίο μελετάμε, το πλαίσιο αναφοράς του αισθητήρα, καθώς και το παγκόσμιο πλαίσιο αναφοράς, δηλαδή το πλαίσιο αναφοράς της γης. Αν ο

αισθητήρας είναι πάνω στο αντικείμενο τότε τα 2 πρώτα πλαίσια είναι ίδια οπότε το πλαίσιο αναφοράς του σώματος δεν μας ενδιαφέρει. Επομένως η θέση περιγράφεται, ως το πλαίσιο αναφοράς του αισθητήρα προς το πλαίσιο αναφοράς της γης.

Οι μετατοπίσεις των αξόνων περιστροφής δίνουν τις γωνίες Euler, οι οποίες υπολογίζονται από μαθηματικούς τύπους. Παρακάτω ακολουθεί εικόνα με τις γωνίες Euler κατά την περιστροφή των αξόνων.



Εικόνα 3.9: Γωνίες Euler

Οι γωνίες Euler μας βοηθάνε πολύ στην κατανόηση των περιστροφών που πραγματοποιείται σε κάθε άξονα, καθώς και στην λειτουργία του γυροσκοπίου και στον τρόπο με τον οποίο λαμβάνουμε τις τιμές που προέρχονται από αυτόν καθώς ο αδρανειακός αισθητήρας περιστρέφεται. Υπάρχει, όμως, ένα βασικό μειονέκτημα στις γωνίες Euler, το οποίο ονομάζεται gimbal lock. Το gimbal lock είναι η αναπόφευκτη απώλεια ενός βαθμού ελευθερίας όταν οι 2 από τους 3 άξονες γίνονται παράλληλοι. Τότε ο μηχανισμός gimbal “κλειδώνεται” μετατρέποντας την περιστροφή σε δυσδιάστατη, καθώς η περιστροφή 2 διαφορετικών αξόνων προκαλεί την ίδια ακριβώς κίνηση. Ο ακριβής προσανατολισμός, στον οποίο πραγματοποιείται το φαινόμενο αυτό, εξαρτάται από τη σειρά πραγματοποίησης των περιστροφών. Το πρόβλημα αυτό, λοιπόν μπορεί να παρακαμφθεί με τα τετραδόνια (quaternions).

3.3.5 Τετραδόνια (quaternions)

Τα τετραδόνια (quaternions) αποτελούν μια μη αντιμεταθετική επέκταση της θεωρίας των μιγαδικών. Εισήχθησαν στο πεδίο της μηχανικής στον τρισδιάστατο χώρο, από τον Ιρλανδό μαθηματικό Γουίλιαμ Ρόουαν Χάμιλτον το 1843. Παρόλο που σαν έννοια τα τετραδόνια είναι πολύ πιο δυσνόητα από τις γωνίες Euler, στους αλγορίθμους των αισθητήρων χρησιμοποιούνται αυτά, καθώς δεν υποφέρουν από το φαινόμενο gimbal lock, το οποίο προαναφέραμε. Θα αναφερθούμε σε κάποιες εισαγωγικές θεωρητικές έννοιες γύρω από τα τετραδόνια, για να κατανοήσουμε το λόγο για τον οποίον χρησιμοποιούνται. Παρόλα αυτά, δεν θα χρειαστεί να εμβαθύνουμε καθώς δεν κάνουμε εμείς υπολογισμούς. Οι αισθητήρες μας υπολογίζουν τις τιμές quaternion και είτε τις παράγουν ως έξοδο, είτε τις μετατρέπουν σε τιμές των γωνιών Euler για να είναι πιο κατανοητές σε εμάς.

Τα τετραδόνια, λοιπόν, στην επιστήμη των μαθηματικών είναι τετραδιάστατοι μιγαδικοί αριθμοί, οι οποίοι μπορούν να χρησιμοποιηθούν για να αντιπροσωπεύσουν τον προσανατολισμό ενός στερεού σώματος, όπως και οι γωνίες Euler. Με άλλα λόγια τα τετραδόνια κωδικοποιούν την περιστροφή από το αδρανειακό πλαίσιο (σύστημα συντεταγμένων με βάση την Γη) στο πλαίσιο του αισθητήρα. Σε αυτό το αδρανειακό πλαίσιο, ο x-άξονας δείχνει προς τα βόρεια, ο y-άξονας δείχνει ανατολικά και ο z-άξονας δείχνει προς τα κάτω. Τα τετραδόνια αποτελούνται από ένα πραγματικό στοιχείο και τρία φανταστικά στοιχεία.

Στο πεδίο των μιγαδικών αριθμών, τα τετραδόνια προκύπτουν από την πρόσθεση των βασικών στοιχείων i , j και k σε πραγματικούς αριθμούς, όπου τα i , j και k ικανοποιούν τη σχέση:

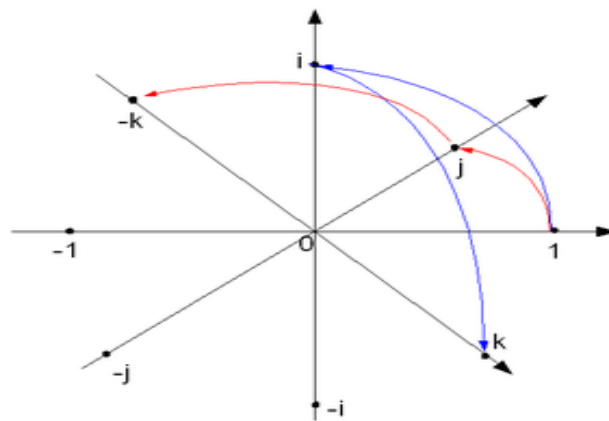
$$i^2 = j^2 = k^2 = ijk = -1$$

Κάθε τετραδόνιο αποτελεί γραμμικό συνδυασμό των βασικών τετραδονίων 1, i , j και k . Έτσι, μπορεί να εκφρασθεί με μοναδικό τρόπο ως $w + x i + y j + z k$ όπου w , x , y , z είναι πραγματικοί αριθμοί. Ο πολλαπλασιασμός είναι προσεταιριστικός, επομένως εάν πολλαπλασιάσουμε το k στη σχέση $-1 = ijk$, έχουμε:

$$\begin{aligned} -k &= ijkk = ij(k^2) = ij(-1), \\ k &= ij. \end{aligned}$$

Αντιστοίχως έχουμε και τις σχέσεις:

$$\begin{aligned} ij &= k, & ji &= -k, \\ jk &= i, & kj &= -i, \\ ki &= j, & ik &= -j, \end{aligned}$$



Graphical representation of quaternion units product as 90°-rotation in 4D-space

$$\begin{aligned} ij &= k \\ ji &= -k \\ ij &= -ji \end{aligned}$$

Εικόνα 3.10: Περιστροφή μοναδιαίων τετραδονίων στον τετραδιάστατο χώρο

Ακολουθούν οι σχέσεις μετατροπής τετραδονίου (q.w, q.x, q.y, q.z) σε γωνίες Euler (yaw, pitch, roll):

```
// roll (x-axis rotation)
sinr = +2.0 * (q.w() * q.x() + q.y() * q.z());
cosr = +1.0 - 2.0 * (q.x() * q.x() + q.y() * q.y());
roll = atan2(sinr, cosr);

// pitch (y-axis rotation)
sinp = +2.0 * (q.w() * q.y() - q.z() * q.x());
if (fabs(sinp) >= 1)
    pitch = copysign(M_PI / 2, sinp);
else
    pitch = asin(sinp);

// yaw (z-axis rotation)
siny = +2.0 * (q.w() * q.z() + q.x() * q.y());
cosy = +1.0 - 2.0 * (q.y() * q.y() + q.z() * q.z());
yaw = atan2(siny, cosy);
```

Για την αντίθετη σχέση, δηλαδή τη μετατροπή από γωνίες Euler σε τετραδόνια, θα έχουμε τις εξής σχέσεις:

```
cy = cos(yaw * 0.5);
sy = sin(yaw * 0.5);
cr = cos(roll * 0.5);
sr = sin(roll * 0.5);
cp = cos(pitch * 0.5);
sp = sin(pitch * 0.5);

q.w() = cy * cr * cp + sy * sr * sp;
q.x() = cy * sr * cp - sy * cr * sp;
q.y() = cy * cr * sp + sy * sr * cp;
q.z() = sy * cr * cp - cy * sr * sp;
```

3.3.6 Σύστημα αναφοράς θέσης στίγματος (AHRS)

Το σύστημα αναφοράς θέσης στίγματος, σε μία ελεύθερη μετάφραση από τον αγγλικό όρο Attitude and Heading Reference System (AHRS), είναι ένα σύστημα αναφοράς το οποίο προσδιορίζει τον προσανατολισμό ενός αντικειμένου στον τρισδιάστατο χώρο. Ουσιαστικά, αυτό το σύστημα αναφοράς αποτελείται από μικροηλεκτρομηχανικά συστήματα (Microelectromechanical System - MEMES) αισθητήρων τριών αξόνων (x, y, z). Ένα τέτοιο ολοκληρωμένο σύστημα παρέχει πληροφορίες MARG (Magnetic - Angular Rate - Gravity), δηλαδή πληροφορίες για το μαγνητικό πεδίο στους 3 άξονες μέσω ενός μαγνητόμετρου (magnetometer), πληροφορίες για το γωνιακό ρυθμό ή αλλιώς τη γωνιακή επιτάχυνση στους 3 άξονες μέσω ενός γυροσκοπίου (gyroscope) και πληροφορίες για τις δυνάμεις επιτάχυνσης λόγω της βαρύτητας στους 3 άξονες μέσω ενός επιταχυνσιόμετρου (accelerometer). Επομένως, ένα τέτοιο ολοκληρωμένο σύστημα που μας παρέχει τις 3 τιμές (x, y, z) από τα 3 μικροηλεκτρομηχανικά συστήματα (μαγνητόμετρο, γυροσκόπιο, επιταχυνσιόμετρο), μπορούμε να το ονομάσουμε αδρανειακό αισθητήρα κίνησης 9 αξόνων (9-axis imu). Τα συστήματα στα οποία λείπουν οι 3 τιμές μαγνητόμετρου τα ονομάζουμε αδρανειακούς αισθητήρες 6 αξόνων (6-axis imu).

3.3.7 Αδρανειακοί αισθητήρες (IMU)

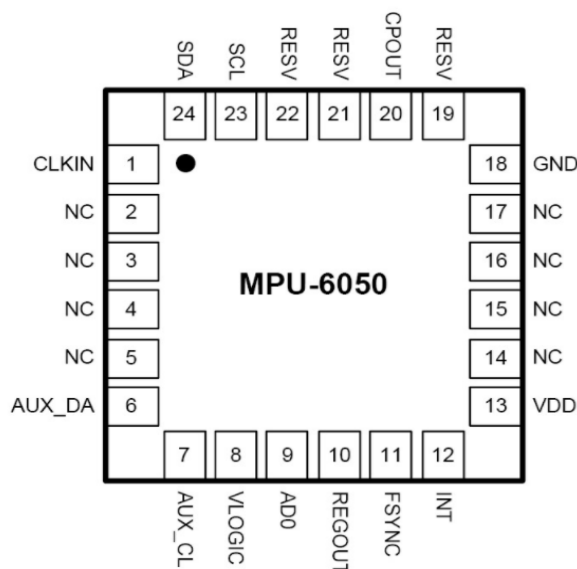
Οι αδρανειακοί αισθητήρες (IMUs) είναι ηλεκτρονικές συσκευές οι οποίες μετρούν την κλίση, την επιτάχυνση και τον προσανατολισμό ενός σώματος στον χώρο, με τη βοήθεια γυροσκοπίων, επιταχυνσιόμετρων και μαγνητοσκοπίων. Οι συσκευές αυτές χρησιμοποιούνται στην αδρανειακή πλοήγηση, σε ελιγμό αεροσκαφών και μη επανδρωμένα εναέρια οχήματα, διαστημικά οχήματα, δορυφόρους, GPS, καθώς και σε διάφορες εφαρμογές έξυπνων κινητών (smartphones) και επιστημονικών εργαλείων μέτρησης. Οι αισθητήρες αυτοί ονομάζονται αδρανειακοί, διότι η κίνησή τους περιλαμβάνεται στο αδρανειακό σύστημα αναφοράς, στο οποίο ισχύουν ο πρώτος και ο δεύτερος νόμος του Νεύτωνα.

3.3.8 Αδρανειακός αισθητήρας κατασκευής

Ο αδρανειακός αισθητήρας (IMU), ο οποίος χρησιμοποιήσα στη συγκεκριμένη πτυχιακή εργασία είναι ο MPU6050. Η πλακέτα κανονικά έχει την ονομασία GY-521 (breakout board), συχνά όμως συναντάται και ως MPU6050 από τον κεντρικό αισθητήρα (MEMS) της InvenSense που διαθέτει, τον MPU6050, του οποίου τη λειτουργία θα αναλύσουμε.

3.3.8.1 MPU6050

Πρόκειται για μια πλακέτα η οποία ενσωματώνει ψηφιακό γυροσκόπιο τριών αξόνων, ψηφιακό επιταχυνσιόμετρο τριών αξόνων καθώς και έναν μικροελεγκτή (DMP), σε ένα chip. Ο ενσωματωμένος μικροελεγκτής λαμβάνει τις μετρήσεις από τα αισθητήρια και εφαρμόζει ψηφιακά φίλτρο DCM (Direction Cosine Matrix) μέσω αλγορίθμων για να δώσει την ακριβή γωνία κλίσης σε κάθε άξονα. Η επικοινωνία μεταξύ αισθητηρίων και μικροελεγκτή επιτυγχάνεται μέσω του πρωτοκόλλου I2C, το οποίο θα αναλύσουμε παρακάτω.

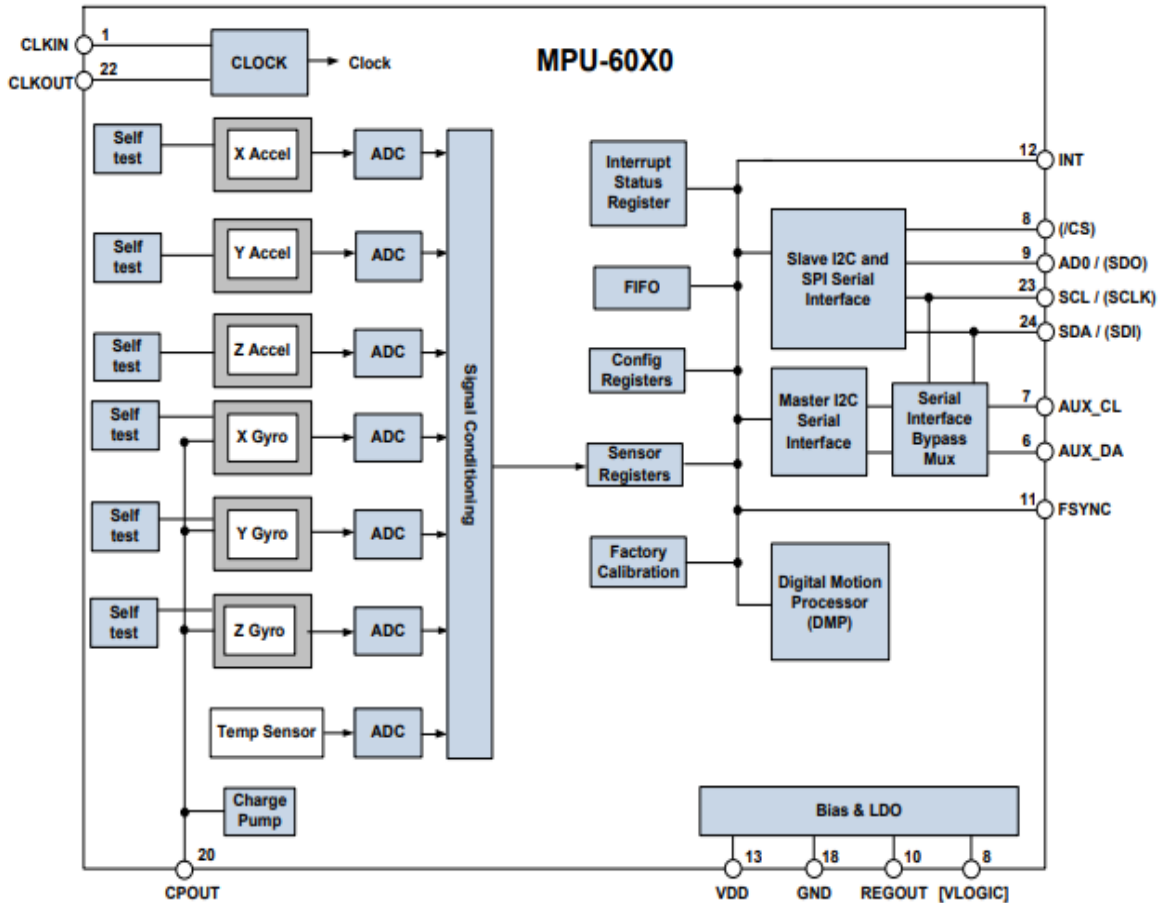


Εικόνα 3.11: MPU6050

Τα χαρακτηριστικά του αισθητήρα MPU6050 είναι τα εξής:

- 3-Axis αισθητήρας γωνιακής ταχύτητας (γυροσκόπιο) με πλήρες φάσμα κλίμακας ± 250 , ± 500 , ± 1000 , και ± 2000 dps (degrees per second) και 16-bit μετατροπέα αναλογικού σήματος σε ψηφιακό (ADC).
- 3-Axis επιταχυνσιόμετρο με πλήρες φάσμα κλίμακας προσαρμοζόμενο σε $\pm 2g$, $\pm 4g$, $\pm 8g$ και $\pm 16g$ και 16-bit μετατροπέα αναλογικού σήματος σε ψηφιακό (ADC).
- Ψηφιακό επεξεργαστή κίνησης (DMP) ο οποίος εκτελεί αλγορίθμους και δίνει τις βέλτιστες τιμές εξόδου (θα αναλυθεί παρακάτω).
- Διεπαφές σειριακού διαύλου I2C (I2C serial communication interface) και διαύλου σειριακής περιφερειακής διασύνδεσης (SPI)
- Βοηθητική Διεπαφή σειριακού διαύλου I2C (auxiliary I2C serial interface) που επικοινωνούν μέσω auxiliary I2C bus, στις οποίες μπορούν να συνδεθούν εξωτερικοί αισθητήρες (external / off-chip), όπως μαγνητόμετρο, και μέσω του I2C master mode να σταλούν οι τιμές στον DMP
- Μετρητές (clocking)
- Καταχωρητές δεδομένων αισθητήρων (sensor data registers)
- Δομή δεδομένων FIFO
- Διακοπές (interrupts), και καταχωρητές διακοπών για συσκευές που χρειάζονται να πυροδοτηθεί (trigger) η έναρξη ή η παύση λειτουργίας τους.
- Ψηφιακή έξοδος αισθητήρα θερμοκρασίας
- Αντλία φόρτισης (charge pump)
- Self-test επιταχυνσιόμετρου και γυροσκοπίου για έλεγχο τιμών τους σε ένα εύρος min-max.
- Τμήμα bias and LDO, το οποίο είναι υπεύθυνο για την εσωτερική παροχή ρευμάτων και τάσεων που απαιτούνται από τον αισθητήρα

Όλα αυτά τα μέρη του chip αυτού λειτουργούν με μία αλληλουχία, ώστε οι τελικές τιμές που παράγονται ως έξοδοι από τον αισθητήρα να είναι όσο το δυνατόν πιο ακριβής . Στο παρακάτω σχήμα παρουσιάζεται ο τρόπος επικοινωνίας των επιμέρους μερών, επομένως και ο τρόπος λειτουργίας του αισθητήρα.



Εικόνα 3.12: Block diagram του MPU6050

3.3.8.2 Δίαυλος I2C

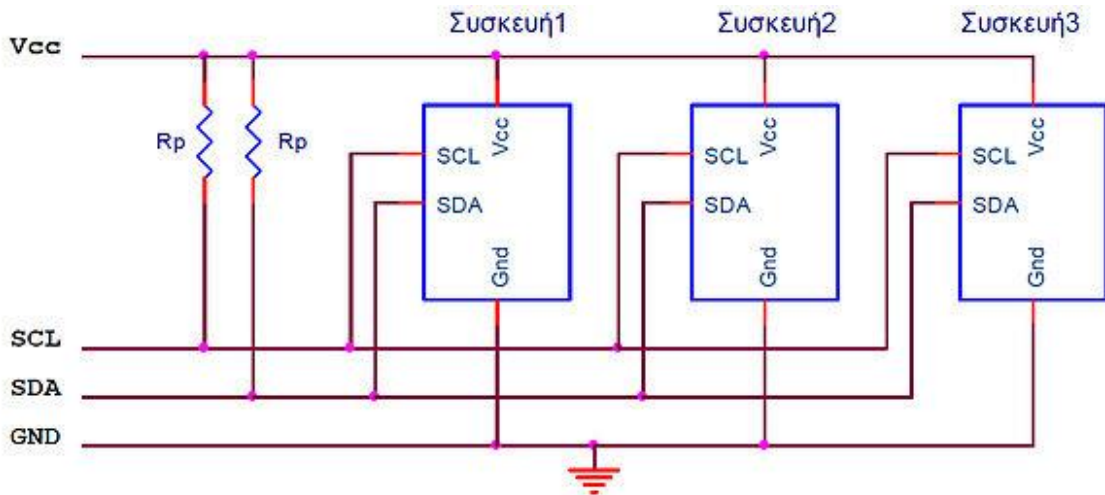
Ο δίαυλος I2C είναι ένας σειριακός δίαυλος που δημιουργήθηκε από τη Philips (τώρα NXP) και χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε μητρικές πλακέτες (motherboards), ενσωματωμένα συστήματα (embedded systems), κινητά τηλέφωνα ή άλλες ηλεκτρονικές συσκευές. Ο δίαυλος αυτός, δεν χρησιμοποιείται μόνο για την επικοινωνία συσκευών που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα, αλλά και για την επικοινωνία συσκευών που συνδέονται με καλώδια.

Στο δίαυλο I2C χρησιμοποιούνται 2 γραμμές για μεταφορά δεδομένων:

- Η γραμμή ρολογιού, SCL (παλμοί ρολογιού)
- Η γραμμή δεδομένων, SDA (bit δεδομένων)

Επίσης χρειάζονται 2 ακόμα γραμμές για τροφοδοσία:

- Η γραμμή γείωσης, GND
- Η γραμμή τροφοδοσίας (Vcc)



Εικόνα 3.13: Δίαυλος I2C

Κάθε bit δεδομένων που μεταφέρεται στη γραμμή SDA συνοδεύεται και από έναν παλμό ρολογιού στη γραμμή SCL. Η μέγιστη συχνότητα λειτουργίας μπορεί να φτάσει τα 400kHz. Στο δίαυλο μπορούν να συνδεθούν μέχρι 128 συσκευές (για τον standard mode) και να λαμβάνουν ή να αποστέλλουν δεδομένα. Οι μοναδικοί περιορισμοί στον αριθμό των συσκευών που μπορούν να συνδεθούν, προκύπτουν από τη συνολική χωρητικότητα που μπορεί να αποκτήσει ο δίαυλος, η οποία δεν μπορεί να ξεπεράσει τα 400pF και από τον περιορισμό των διευθύνσεων, ο οποίος περιορίζεται στα 7 bits. Σε όλες τις slave συσκευές που συνδέονται στον δίαυλο, έχει αποδοθεί ένας αριθμός σαν διεύθυνση (address).

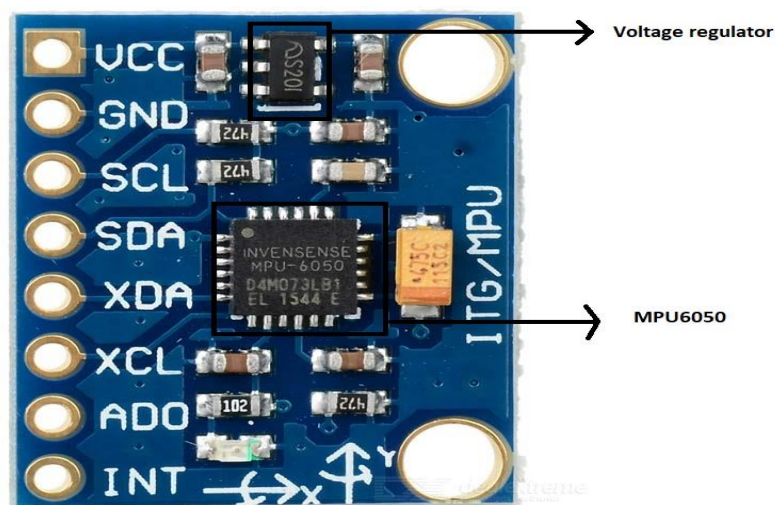
Το I2C πρωτόκολλο (I2C software protocol), με το οποίο πραγματοποιείται η I2C επικοινωνία, εκτελεί αμφίδρομη επικοινωνία από την master συσκευή στην/στις slave και από την/τις slave στην master, εφόσον είναι γνωστές οι διευθύνσεις των slave, στέλνοντας πακέτα δεδομένων στις γραμμές SDA, SCL.

3.3.8.3 DMP

Ο DMP (Digital Motion Processor), είναι ένας ενσωματωμένος ψηφιακός επεξεργαστής κίνησης, που βρίσκεται εντός του MPU6050, και κάνει υπολογισμούς για φόρτωση των αλγορίθμων επεξεργασίας κίνησης από τον κεντρικό επεξεργαστή. Ο DMP, παίρνει δεδομένα από γυροσκόπιο, επιταχυνσιόμετρο, καθώς και εξωτερικά συνδεδεμένους αισθητήρες (auxiliary) όπως το μαγνητόμετρο και επεξεργάζεται αυτά τα δεδομένα. Ο DMP διαθέτει καταχωρητές για να διαβάσει τα δεδομένα και FIFO (ουρές) για να τα αποθηκεύσει. Επίσης έχει πρόσβαση σε μία εξωτερική ακίδα του MPU6050, για τη δημιουργία και διαχείριση διακοπών (interrupts). Σκοπός του DMP, είναι να βελτιώσει τις απαιτήσεις χρονισμού και την ισχύ επεξεργασίας του κεντρικού επεξεργαστή. Τυπικά, οι αλγόριθμοι επεξεργασίας κίνησης θα πρέπει να εκτελούνται με υψηλό ρυθμό (γύρω στα 200Hz) για να παρέχουν ακριβή αποτελέσματα με χαμηλό χρόνο απόκρισης (low latency).

3.3.8.4 Πλακέτα GY-521

Η πλακέτα η οποία χρησιμοποιήθηκε στην εργασία, για την ανίχνευση κίνησης, είναι η GY-521 ή αλλιώς MPU6050, καθώς το κεντρικό chip της πλακέτας, το οποίο πραγματοποιεί όλους τους υπολογισμούς που χρειαζόμαστε είναι ο MPU6050, που περιγράψαμε παραπάνω.



Εικόνα 3.14: GY-521

Η πλακέτα αυτή διαθέτει 8 ακροδέκτες. Οι 4 πρώτοι είναι γνωστοί από την I2C επικοινωνίας που περιγράψαμε προηγουμένως, δηλαδή: ακροδέκτης Vcc τροφοδοσίας (5V), ακροδέκτης γείωσης GND (0V), ακροδέκτης SCL γραμμής ρολογιού για I2C επικοινωνία και ακροδέκτης SDA γραμμής δεδομένων για I2C επικοινωνία.

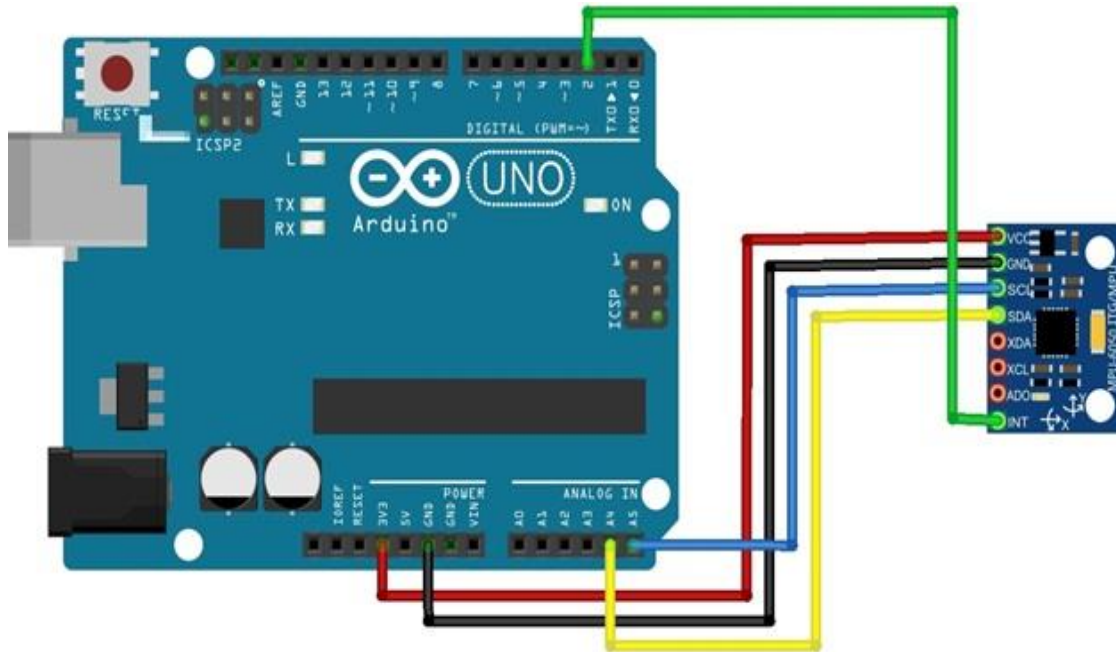
Οι επόμενοι 3 ακροδέκτες (XDA, XCL, ADO) δεν χρησιμοποιούνται συχνά (δεν θα χρειαστούν στην παρούσα εργασία). Οι 2 πρώτοι (XDA, XCL) είναι για τη βοηθητική γραμμή I2C (auxiliary I2C bus) και χρησιμοποιούνται στην περίπτωση που θα θέλαμε να συνδέσουμε άλλους αισθητήρες (πχ μαγνητόμετρο). Ο ακροδέκτης ADO χρησιμοποιείται για την ίδια περίπτωση και είναι αυτός που μας δίνει τη δυνατότητα να επιλέξουμε μεταξύ των I2C διευθύνσεων 0x68 και 0x69 (αυτές είναι οι δυνατές I2C διευθύνσεις αν χρησιμοποιούμε μία γραμμή I2C), άρα μπορούμε να έχουμε 2 αισθητήρες σε αυτή τη γραμμή (για παραπάνω αισθητήρες χρειαζόμαστε πολυπλέκτη multiplexer).

Ο τελευταίος ακροδέκτης είναι ο INT, αυτός δηλαδή που είναι υπεύθυνος για τις διακοπές (interrupts). Με αυτό τον ακροδέκτη, ο μικροεπεξεργαστής καταλαβαίνει ότι έχουν φτάσει νέα δεδομένα στους καταχωρητές και έτσι μπορούν να διαβαστούν πια από αυτόν. Επίσης, η πρώτη διακοπή, πυροδοτεί (trigger) τον αισθητήρα μας για να ξεκινήσει να λειτουργεί, καθώς πριν αυτό συμβεί βρίσκεται σε sleep mode.

3.3.8.5 Βαθμονόμηση αισθητήρα κίνησης

Ο αισθητήρας κίνησης για να δώσει σωστές μετρήσεις θα πρέπει να βαθμονομηθεί. Η διαδικασία βαθμονόμησης του αισθητήρα ονομάζεται calibration και είναι η διαδικασία με την οποία εντοπίζουμε τα αντισταθμίσματα ή αλλιώς offsets του αισθητήρα μας. Κάθε αισθητήρας έχει διαφορετικά offsets, τα οποία είναι 6 τιμές (3 τιμές γυροσκοπίου gyroX, gyroY, gyroZ και 3 τιμές επιταχυνσιόμετρου accelX, accelY, accelZ). Γνωρίζοντας λοιπόν αυτές τις 6 τιμές μπορούμε να βαθμονομήσουμε το γυροσκόπιο και το επιταχυνσιόμετρο του αισθητήρα μας, κάτι το οποίο απαιτείται για σωστές μετρήσεις.

Η διαδικασία βαθμονόμησης είναι απλή. Για πιο άμεση και εύκολη βαθμονόμηση συνδέουμε σε ένα Arduino Uno τον αισθητήρα μας (MPU6050) ως εξής:



Εικόνα 3.15: Σύνδεση Arduino Uno με MPU6050

Στο διαδίκτυο υπάρχει ελεύθερος ο κώδικας για βαθμονόμηση του αισθητήρα με την ονομασία του sketch “MPU6050_calibration.ino” στο url: <https://forum.arduino.cc/index.php?action=dlattach;topic=446713.0;attach=193816>

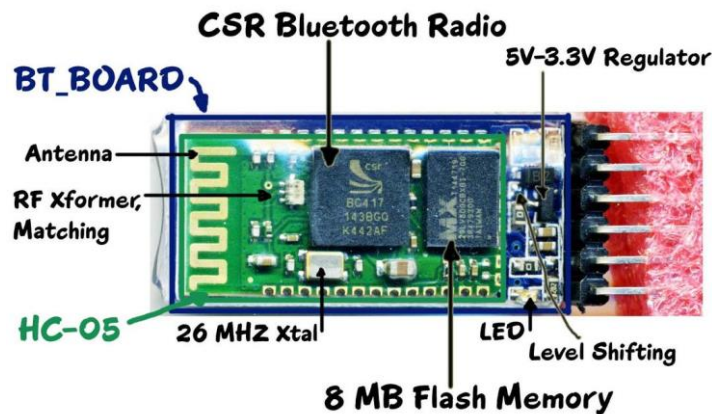
Όπως αναφέρεται και στον κώδικα, ο αισθητήρας θα πρέπει να βρίσκεται σε οριζόντια θέση με τα γράμματα να κοιτάνε προς τα πάνω και να μείνει ακίνητος μέχρι να βρεθούν τα offsets. Μετά από λίγα δευτερόλεπτα αφού ξεκινήσει το πρόγραμμα να τρέχει και να υπολογίζει μας επιστρέφει 6 τιμές, οι οποίες είναι τα offset του αισθητήρα μας. Αυτές τις τιμές τις σημειώνουμε κάπου, ώστε να τις έχουμε όταν θα χρησιμοποιήσουμε τον αισθητήρα και θα πρέπει να δηλώσουμε τα offset του για να λειτουργήσει σωστά.

3.4 Ασύρματη επικοινωνία

Σε αυτό το υποκεφάλαιο θα περιγράψουμε την τεχνολογία, την οποία χρησιμοποιήσαμε για να πετύχουμε την ασύρματη επικοινωνία του κυκλώματός μας με τον υπολογιστή.

3.4.1 HC-05

Το bluetooth module HC-05, είναι ένα Bluetooth SPP (Serial Port Protocol) module εύκολο σε χρήση. Με άλλα λόγια είναι μία πλακέτα για ασύρματη σειριακή επικοινωνία και μπορεί να χρησιμοποιηθεί είτε ως master είτε ως slave.



Εικόνα 3.16: Επιμέρους τμήματα πλακέτας HC-05

Χρησιμοποιεί την έκδοση Bluetooth 2.0 + EDR, η οποία εμφανίστηκε το 2004. Ο ρυθμός μεταφοράς δεδομένων είναι 3 Mbps, με 2.4GHz ραδιοπομποδέκτη και ζώνη βάσης. Επίσης, χρησιμοποιεί το chip CSR Bluecore 04.

Χαρακτηριστικά υλικού

- -80dBm ευαισθησία
- Πάνω από +4dBm δύναμη μετάδοσης ραδιοσυχνότητας (RF)
- 3.3 έως 5V είσοδος/έξοδος (I/O)
- Έλεγχος PIO(Programmable Input/Output)
- Διεπαφή UART με προγραμματιζόμενη ταχύτητα μετάδοσης
- Ενσωματωμένη κεραία

Χαρακτηριστικά λογισμικού

- Αυτόματες ρυθμίσεις συσκευής (Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity)
- Αυτόματη σύνδεση στην τελευταία συνδεδεμένη συσκευή
- Επιτρέπεται η σύνδεση με ζεύξη συσκευών ως προεπιλογή
- Κωδικός ζεύξης: "1234" ως προεπιλογή



Εικόνα 3.17: Ακροδέκτες HC-05

Το bluetooth module HC-05, διαθέτει 6 ακροδέκτες. Ο ακροδέκτης state, είναι δείκτης κατάστασης. Αν η συσκευή είναι συνδεδεμένη, με άλλη συσκευή bluetooth το σήμα είναι High, αλλιώς είναι Low.

Οι 2 επόμενοι ακροδέκτες (RXD, TXD) χρησιμοποιούνται για επικοινωνία σαν UART interface. Ο ακροδέκτης RXD (Receive Serial Data), λαμβάνει δεδομένα από το TXD (transmit data) του Arduino μας, ενώ αντίθετα ο ακροδέκτης TXD (Transmit Serial Data), στέλνει δεδομένα στο RXD (receive data) του Arduino.

Οι 2 αμέσως επόμενοι ακροδέκτες (GND, VCC) είναι οι γνωστοί πια ακροδέκτες τροφοδοσίας VCC (5V) και γείωσης GND (0V).

Ο τελευταίος ακροδέκτης είναι ο EN (enable) και αν το σήμα εδώ γίνει LOW, η συσκευή απενεργοποιείται, δεν ανοίγει και δεν επιτρέπεται η επικοινωνία. Αν το σήμα παραμείνει HIGH ή συνδεθεί με 3.3V η συσκευή διατηρείται ανοιχτή και η επικοινωνία πραγματοποιείται κανονικά.

Τέλος, υπάρχει και ένα κουμπί (switch button), το οποίο αν πατηθεί παρατεταμένα θέτει τη συσκευή σε λειτουργία εντολών κι έτσι ο χρήστης μπορεί να αλλάξει ορισμένες παραμέτρους της συσκευής. Αυτό δεν μπορεί να συμβεί μόνο στην περίπτωση που η συσκευή είναι συνδεδεμένη με άλλη bluetooth συσκευή. Οι ακροδέκτες, οι οποίοι θα χρειαστούν στην εργασία μας είναι οι 2 της τροφοδοσίας (VCC, GND), οι 2 της επικοινωνίας (RXD, TXD) και αυτός των διακοπών (INT).

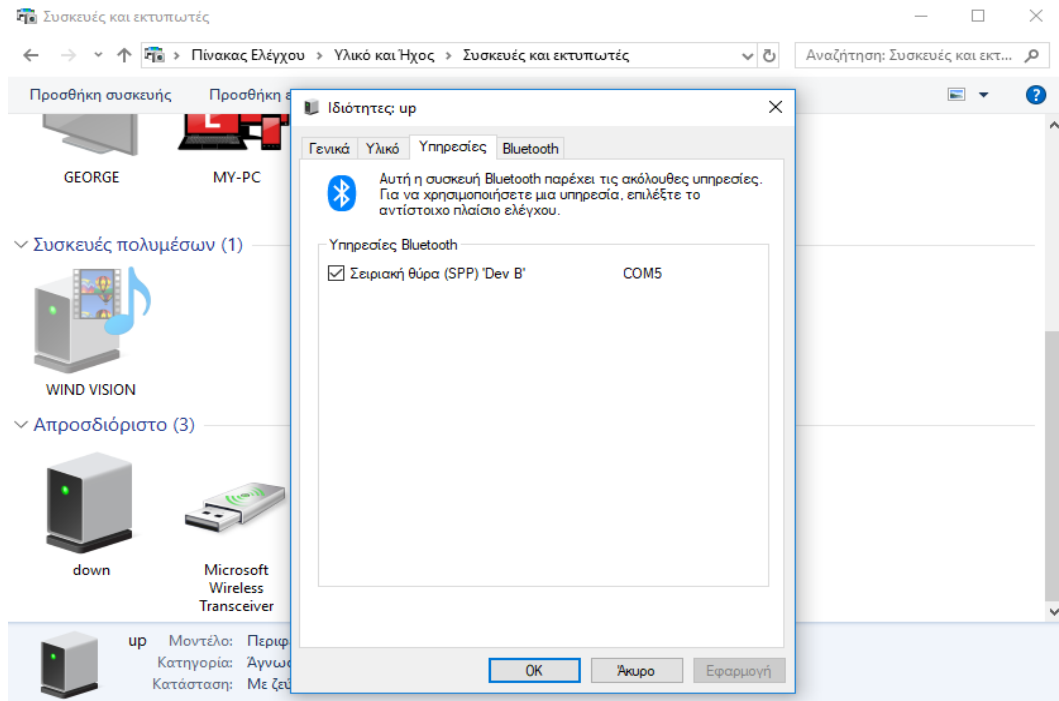
3.4.2 Επικοινωνία συσκευής με σειριακή πόρτα

Όταν μία συσκευή bluetooth, όπως αυτή που χρησιμοποιούμε ανιχνευτεί από το bluetooth του υπολογιστή μας και πραγματοποιήσουμε μια ζεύξη με αυτή τη συσκευή (χρησιμοποιώντας τον προκαθορισμένο κωδικό: "1234"), αυτόματα δημιουργούνται στον υπολογιστή μας δύο ασύρματες σειριακές πόρτες (Bluetooth SPP Driver COMX), οι οποίες επικοινωνούν με τη συσκευή. Η μία δέχεται δεδομένα και η άλλη στέλνει δεδομένα (Incoming-outgoing). Μπορούμε από τον πίνακα ελέγχου να βρούμε την bluetooth συσκευή μας (εμφανίζεται με προκαθορισμένο όνομα HC-05) και ανοίγοντας τις ιδιότητες μπορούμε να βρούμε με ποια σειριακή πόρτα επικοινωνεί (εικόνα 3.18), να αλλάξουμε την πόρτα επικοινωνίας, της ιδιότητες της πόρτας ή της συσκευής, καθώς και το όνομα της. Επίσης οι πόρτες που δημιουργούνται κατά την ζεύξη της bluetooth συσκευής με τον υπολογιστή, εμφανίζονται στη διαχείριση συσκευών των Windows (εικόνα 3.19). Ανοίγοντας της ιδιότητες της κάθε πόρτας μπορούμε να αλλάξουμε τις προκαθορισμένες ρυθμίσεις.

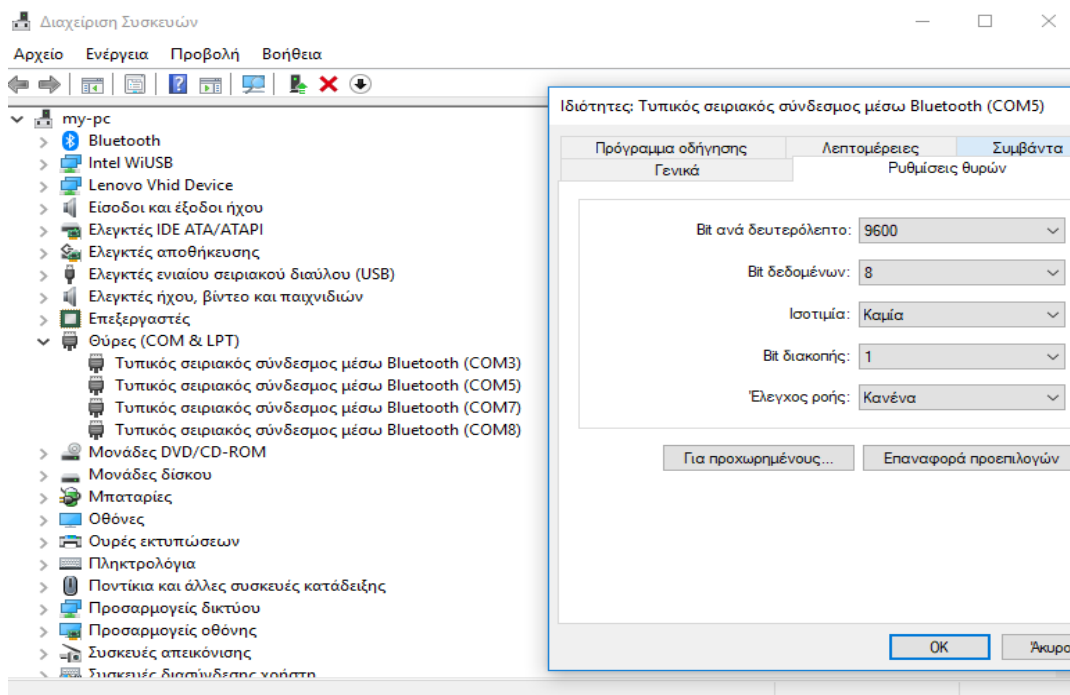
Η μοναδική ρύθμιση που χρειαζόμαστε για την εργασία, εφόσον έχουμε 2 συσκευές HC-05 (μία σε κάθε κύκλωμα), είναι να πραγματοποιήσουμε ζεύξη με την πρώτη συσκευή (βάζοντας τον κωδικό: "1234"), να τις αλλάξουμε ονομασία διότι θα έχει το ίδιο όνομα με την άλλη συσκευή και θα μας μπερδέψει και τέλος να σημειώσουμε σε ποια πόρτα στέλνει τα δεδομένα η συσκευή. Έπειτα, κάνουμε ακριβώς το ίδιο και για τη δεύτερη συσκευή. Έτσι έχουμε 2 bluetooth συσκευές με διαφορετικές ονομασίες (στην εργασία η μία λέγεται up και η άλλη down) και 2 σειριακές πόρτες που επικοινωνούν με αυτές τις συσκευές. Τέλος, σημειώνουμε ποια συσκευή επικοινωνεί με ποια πόρτα, καθώς όταν χρειαστεί να διαβάσουμε τις τιμές κάθε κυκλώματος θα πρέπει να ανοίξουμε και να διαβάσουμε τη σωστή

Ασύρματη καταγραφή κίνησης χεριού

σειριακή πόρτα. Στην εργασία μας η συσκευή up στέλνει τιμές στη σειριακή πόρτα COM5, ενώ η συσκευή down στέλνει τιμές στη σειριακή πόρτα COM8.



Εικόνα 3.18: Ιδιότητες συσκευής HC-05 στον πίνακα ελέγχου των Windows



Εικόνα 3.19: Ιδιότητες σειριακής πόρτας στη διαχείριση συσκευών των Windows

3.5 Παρουσίαση κατασκευής

Εφόσον είδαμε ένα-ένα ξεχωριστά τα μέρη, τα οποία συνέθεσαν την τελική κατασκευή μας, σε αυτό το υποκεφάλαιο θα δούμε πως όλα αυτά συνδέονται μεταξύ τους, θα γίνει η παρουσίαση της κατασκευής, ο κώδικας που γράψαμε για να την προγραμματίσουμε και η ολοκλήρωσή της για την τελική μορφή.

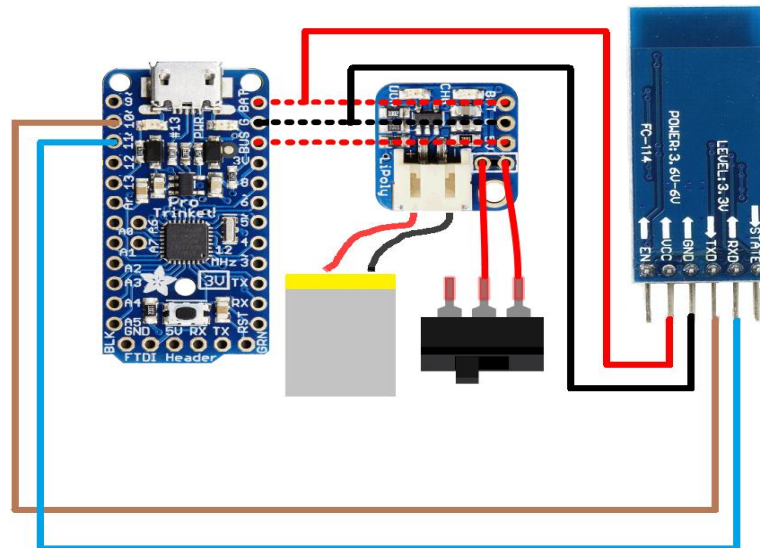
3.5.1 Εισαγωγή στην κατασκευή

Οι τεχνολογίες οι οποίες προαναφέρθηκαν και αναλύθηκαν εκτενώς στα προηγούμενα κεφάλαια, ήταν αυτές που χρησιμοποιήθηκαν για την υλοποίηση της κατασκευής. Ανακεφαλαιώνοντας, καθένα από τα 2 ίδια κυκλώματα διαθέτει:

- Κεντρική πλακέτα → Adafruit Pro Trinket 5V- 16mhz
- Συσσκευή bluetooth για ασύρματη επικοινωνία → HC-05
- Αδρανειακό αισθητήρα για ανίχνευση κίνησης → MPU6050
- Πλακέτα φόρτισης μπαταρίας → Adafruit Pro Trinket Liion/Lipoly Backpack Add-On
- Μπαταρία → Li-Po 3.7V – 250mAh
- Διακόπτης on/off → Slide Switch Mini SPDT 0.3A/30VDC

Η ιδέα της τελικής μορφής της κατασκευής ήταν 2 πανομοιότυπα κυκλώματα, με τη σύνδεση όλων των παραπάνω υλικών, τα οποία αφού θα συνδέονταν με τον καλύτερο πιθανό τρόπο για εξοικονόμηση χώρου, θα κλείνονταν σε κουτάκια τα οποία θα είχαν μία τρύπα για ένα διακόπτη on/off και μία τρύπα για την usb υποδοχή ώστε να ανεβάζουμε κώδικα στην κεντρική πλακέτα, καθώς και να φορτίζουμε τη μπαταρία του κυκλώματός μας.

Λόγω του ότι οι 3d εκτυπώσεις κουτιών ήταν σχετικά ακριβές, αποφάσισα να προχωρήσω σε μία πιο DIY κατασκευή. Βρήκα στο διαδίκτυο μικρά κουτάκια που να μπορούν να χωρέσουν όλα τα υλικά του κυκλώματος, αγόρασα 2 ίδια και άνοιξα 2 τρύπες στο καθένα, ώστε να περνά ο διακόπτης και η υποδοχή του usb μέσα από αυτές.

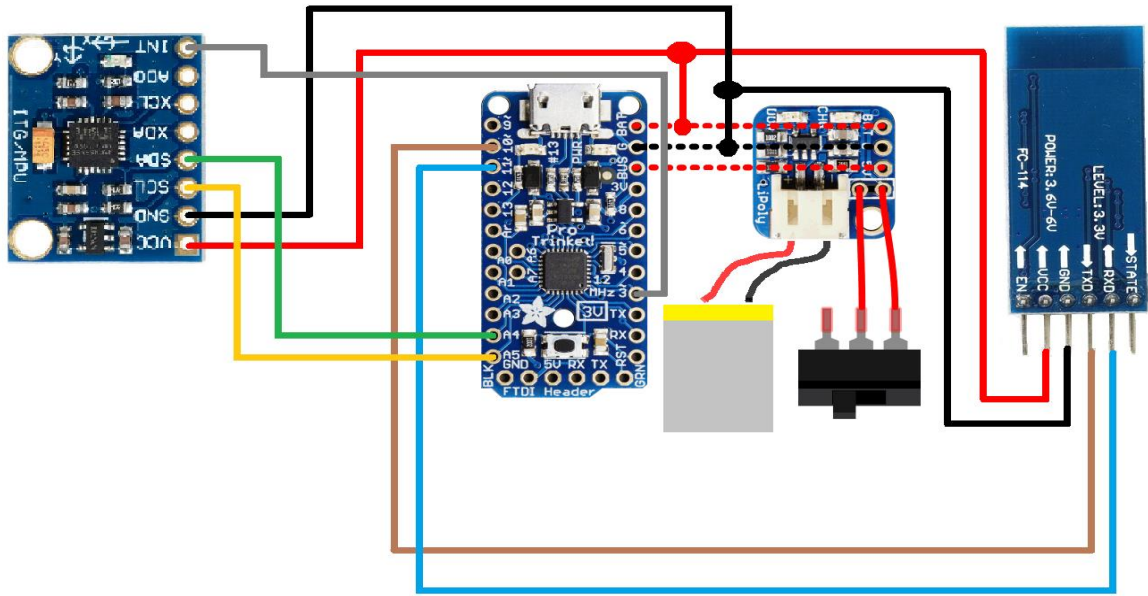


Εικόνα 3.21: Κύκλωμα με προσθήκη της bluetooth συσκευής HC-05

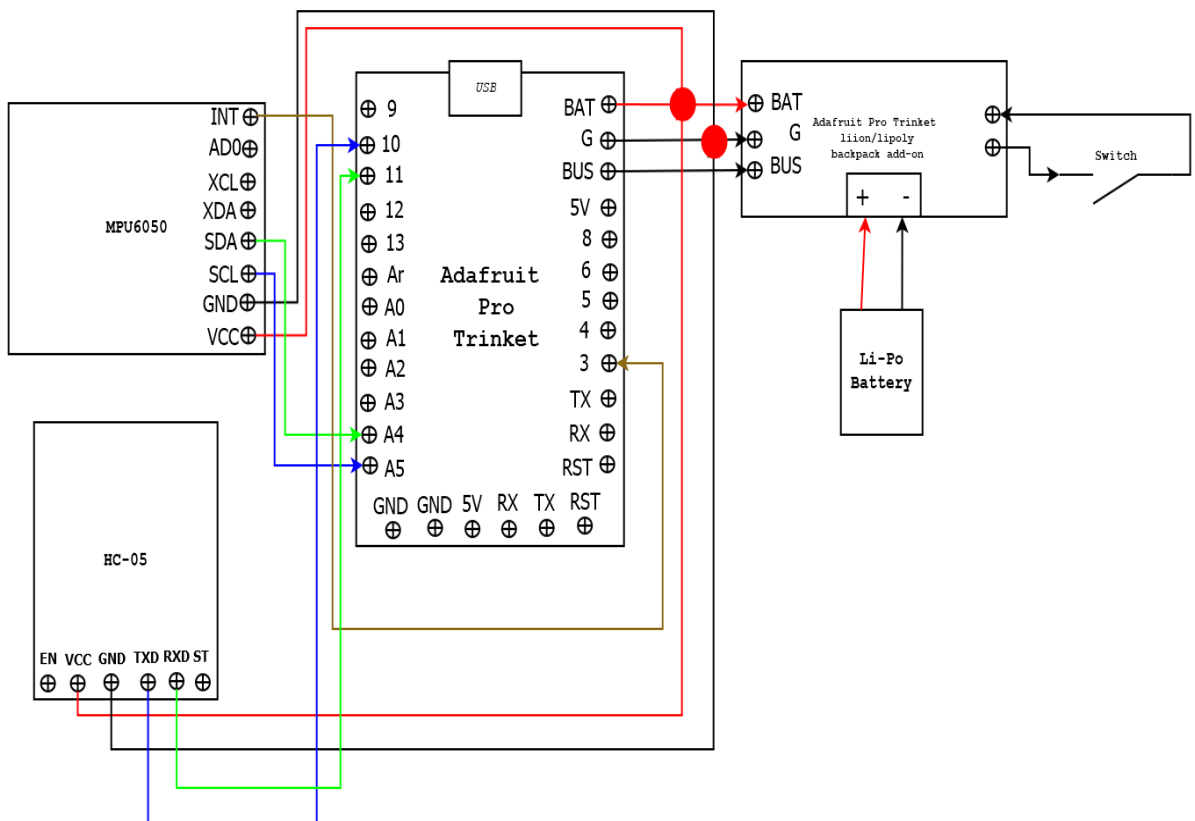
Το τελευταίο βήμα στη συνδεσμολογία του κυκλώματός μου είναι η προσθήκη του αδρανειακού αισθητήρα MPU6050, για την ανίχνευση της κίνησης. Εδώ πρέπει να επισημανθεί ότι προτού ο αισθητήρας συνδεθεί με το υπόλοιπο κύκλωμα, θα πρέπει να έχει περάσει τη διαδικασία βαθμονόμησης με τον τρόπο που περιγράψαμε σε προηγούμενο κεφάλαιο ώστε να γνωρίζουμε τα offset του, τιμές τα οποίες θα χρησιμοποιήσουμε στον κώδικα, με τον οποίο θα προγραμματίσουμε την κεντρική μας πλακέτα. Το ίδιο ισχύει και για τον αισθητήρα του δεύτερου κυκλώματος, ο οποίος θα έχει διαφορετικά offset. Σημαντικό, λοιπόν είναι να γνωρίζουμε ποια offset ανήκουν σε ποιόν αισθητήρα.

Αφού πραγματοποιηθεί και η βαθμονόμηση του αισθητήρα, μπορούμε να τον συνδέσουμε με το υπόλοιπο κύκλωμα. Οι 2 γνωστοί ακροδέκτες τροφοδοσίας του αισθητήρα Vcc και GND θα συνδέονται με τους ακροδέκτες των πλακετών μας BAT και G αντίστοιχα, όπως και η συσκευή HC-05. Οι 2 επόμενοι ακροδέκτες του αισθητήρα, που χρησιμοποιούνται για I2C επικοινωνία, SCL και SDA συνδέονται με τους ακροδέκτες A5 και A4 της κεντρικής μας πλακέτας Adafruit Pro Trinket αντίστοιχα, εφόσον εκεί βρίσκονται οι I2C γραμμές του ρολογιού και των δεδομένων. Τέλος ο ακροδέκτης INT του αισθητήρα που είναι υπεύθυνος για τις διακοπές και την ενεργοποίηση του αισθητήρα, συνδέεται με τον ακροδέκτη 3 της κεντρικής μας πλακέτας Adafruit Pro Trinket, αφού αυτός ο ακροδέκτης ορίζεται ως INT1 (interrupt) από τους κατασκευαστές της.

Ασύρματη καταγραφή κίνησης χεριού



Εικόνα 3.22: Συνδεσμολογία τελικού κυκλώματος



Εικόνα 3.23: Block diagram τελικού κυκλώματος

3.5.3 Ολοκλήρωση κατασκευής

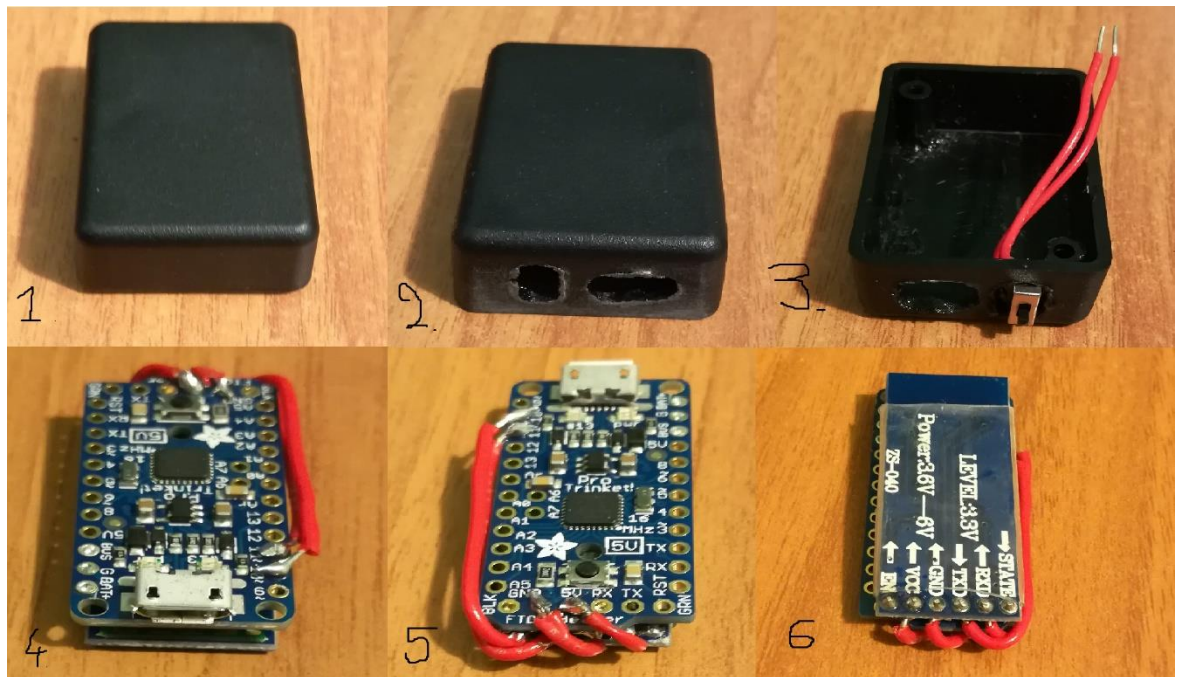
Η τελική συνδεσμολογία του κυκλώματος είναι αυτή που φαίνεται παραπάνω (εικόνα 3.22 και 3.23). Ο τρόπος με τον οποίο θα τοποθετηθούν ώστε να χωρέσουν στο κουτάκι δεν μας απασχολεί, από τη στιγμή που η συνδεσμολογία είναι σωστή. Επομένως, οι πλακέτες μπήκαν κάπως κοντά η μια με την άλλη, ώστε να χωρέσουν στο μέγεθος του κουτιού που αγόρασα.

Στο κουτί απαιτούνται 2 τρύπες, μία για να περνάει ο διακόπτης με τον οποίο θα ανοιγοκλείνουμε τη συσκευή μας και μία για την υποδοχή usb με την οποία θα την φορτίζουμε ή προγραμματίζουμε. Υπολογίζοντας το μέγεθος των τρυπών που έπρεπε να γίνουν πήρα ένα τρυπάνι και άνοιξα τις 2 τρύπες. Μόλις τελείωσα την κατασκευή, κάλυψα με μαύρο στόκο κενά που υπήρχαν στις 2 τρύπες, για καθαρά αισθητικούς λόγους.

Το κουτί είναι κατασκευή της εταιρείας Hammond και έχει διαστάσεις: 50mm x 35mm x 17mm. Έχει μαύρο χρώμα και είναι φτιαγμένο από το υλικό πλαστικό ABS. Η μία πλευρά του αφαιρείται για να μπορέσουμε να τοποθετήσουμε μέσα το κύκλωμά μας και έπειτα κλείνουμε το καπάκι και το βιδώνουμε με 2 βίδες που διαθέτει το κουτί. Έτσι εξασφαλίζουμε ότι η κατασκευή μας δεν θα ανοίξει.

Καθώς σκοπός της πτυχιακής είναι η καταγραφή την κίνηση του χεριού, τα 2 κουτάκια θα έπρεπε να στερεώνονται πάνω στο χέρι, στα 2 σημεία στα οποία ήθελα να ανιχνεύεται η κίνηση. Έτσι έφτιαξα 2 λαστιχένια (για να τεντώνουν) περιβραχιόνια που έχουν ρυθμιζόμενο μήκος (μέσω αυτοκόλλητων). Στην εξωτερική τους πλευρά κόλλησα επίσης ένα αυτοκόλλητο, του οποίου την άλλη μεριά κόλλησα πάνω στο κάθε κουτάκι. Έτσι φορώντας τα περιβραχιόνια στα 2 σημεία του χεριού, στα οποία μετράμε την κίνηση, μπορούμε εύκολα να κολλήσουμε το κουτάκι και να ξεκινήσουμε τις μετρήσεις μας. Το ίδιο εύκολα μπορούμε να το αφαιρέσουμε, κι έτσι γλιτώνουμε τη χρήση κάποιου επιπλέον wearable γαντιού.

3.5.4 Εικόνες κατασκευής



Εικόνα 3.24: Συνδέσεις εξαρτημάτων και διάτρηση κουτιού



Εικόνα 3.25: Τοποθέτηση κυκλώματος στο κουτί και τελική μορφή παρουσίασης

3.5.5 Επεξήγηση κώδικα κατασκευής

Στο διαδίκτυο υπάρχουν πολλοί κώδικες για προγραμματισμό πλακετών Arduino. Ένας κώδικας που είναι γραμμένος για να παρέχει ένα ολοκληρωμένο πακέτο μετρήσεων από τον αισθητήρα που χρησιμοποιούμε είναι ο DMP6. Ονομάζεται MPU6050_DMP6, είναι γραμμένος από τον Jeff Rowberg και βρίσκεται ελεύθερος στη σελίδα:

https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050/examples/MPU6050_DMP6

Για χρήση αυτού του κώδικα θα πρέπει να έχουμε κατεβάσει βιβλιοθήκες (libraries) που να επιτρέπουν εντολές για I2C επικοινωνία. Οι βιβλιοθήκες αυτές βρίσκονται ελεύθερα στο διαδίκτυο, με την ονομασία I2Cdev, στην σελίδα:

<https://github.com/jrowberg/i2cdevlib>

Στην αρχή του κώδικα μπορούμε έπειτα από αυτό να συμπεριλάβουμε τη βιβλιοθήκη με τις εντολές:

```
#include "I2Cdev.h"  
#include "Wire.h"
```

Παρόλα αυτά, ο DMP6 κώδικας αυτός είναι κατάλληλος για ενσύρματη επικοινωνία MPU6050 με Arduino Uno. Εμείς χρειαζόμαστε κώδικα για ασύρματη επικοινωνία, επομένως θα πρέπει να πραγματοποιήσουμε αρκετές αλλαγές ώστε τα αποτελέσματα να μεταδίδονται μέσω του bluetooth module HC-05.

Οι αλλαγές λοιπόν, οι οποίες θα πραγματοποιηθούν στον κώδικα DMP6, για τον προγραμματισμό των συσκευών μας είναι οι παρακάτω:

Προσθέτουμε στην αρχή του κώδικα βιβλιοθήκες σειριακής επικοινωνίας και ορίζουμε ρυθμό μετάδοσης της συσκευής bluetooth αντίστοιχα:

```
#include <SoftwareSerial.h>  
#define BLUETOOTH_SPEED 9600
```

Έπειτα δηλώνουμε τους ακροδέκτες της ασύρματης σειριακής επικοινωνίας :

```
SoftwareSerial mySerial(10,11); //rx,tx
```

Ασύρματη καταγραφή κίνησης χεριού

Επιπλέον αλλάζουμε στην συνάρτηση `setup()` το baud rate της επικοινωνίας από 115200 σε 9600 για να μπορεί να συγχρονιστεί σωστά με το baud rate του bluetooth module HC-05:

```
Serial.begin(9600);  
mySerial.begin(9600);
```

Στη συνέχεια, αλλάζουμε τη συνάρτηση `attachInterrupt`, καθώς η πλακέτα μου Adafruit Pro Trinket έχει τον ακροδέκτη 3 (pin 3) για τα interrupts (INT1), σε αντίθεση με τον Uno που είναι στον ακροδέκτη 2. Έτσι έχω:

```
attachInterrupt(1, dmpDataReady, RISING);
```

Τέλος μέσα στη συνάρτηση `setup()`, θα πρέπει να ορίσουμε τα `offset`, τα οποία πήραμε στη διαδικασία της βαθμονόμησης του αισθητήρα. Κάθε αισθητήρας έχει 6 `offset` τιμές και εφόσον έχω 2 αισθητήρες, θα έχω 12 `offset`. Επομένως σύμφωνα με τις μετρήσεις μου προσθέτω τον κώδικα:

```
//MPU1 offsets  
Mpu.setXGyroOffset(-73);  
Mpu.setYGyroOffset(-10);  
Mpu.setZGyroOffset(14);  
Mpu.setXAccelOffset(-656);  
Mpu.setYAccelOffset(50);  
Mpu.setZAccelOffset(774);  
  
//MPU2 offsets  
Mpu.setXGyroOffset(42);  
Mpu.setYGyroOffset(64);  
Mpu.setZGyroOffset(49);  
Mpu.setXAccelOffset(430);  
Mpu.setYAccelOffset(-607);  
Mpu.setZAccelOffset(1450);
```


Τα πλεονεκτήματα των τετραδονίων με ώθησαν στο να επιλέξω αυτά σαν έξοδο των κυκλωμάτων μου, ώστε να επεξεργαστώ αυτές τις τιμές. Παρόλα αυτά δεν χρησιμοποίησα την συνάρτηση του κώδικα DMP6, δηλαδή την (OUTPUT_READABLE_QUATERNION), λόγω του ότι παρουσιάζει δυσκολίες στην ασύρματη μετάδοση και επεξεργασία. Επέλεξα στη συνάρτηση που δίνει τις γωνίες yaw-pitch-roll, δηλαδή την (OUTPUT_READABLE_YAWPITCHROLL), να προσθέσω τον κώδικα μετατροπής yaw-pitch-roll σε quaternion. Έτσι η συνάρτηση OUTPUT_READABLE_YAWPITCHROLL, την οποία άφησα ως μόνη ενεργή (τοποθέτησα σύμβολα σχολιασμού στις υπόλοιπες) για να παίρνω μόνο τα αποτελέσματα αυτής, μετατράπηκε ως εξής:

```
#ifdef OUTPUT_READABLE_YAWPITCHROLL
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

    cy = cos(ypr[0] *0.5);
    sy = sin(ypr[0] *0.5);
    cr = cos(ypr[2] *0.5);
    sr = sin(ypr[2] *0.5);
    cp = cos(ypr[1] *0.5);
    sp = sin(ypr[1] *0.5);

    mySerial.print("\t");
    mySerial.print(cy * cr * cp + sy * sr * sp);//q.w
    mySerial.print("\t");
    mySerial.print(cy * sr * cp - sy * cr * sp);//q.x
    mySerial.print("\t");
    mySerial.print(cy * cr * sp + sy * sr * cp);//q.y
    mySerial.print("\t");
    mySerial.println(sy * cr * cp - cy * sr * sp);//q.z
    delay(100);
#endif
```

ΚΕΦΑΛΑΙΟ 4

ΑΠΕΙΚΟΝΙΣΗ ΚΙΝΗΣΗΣ ΧΕΡΙΟΥ ΣΕ 3D ΜΟΝΤΕΛΟ

Σε αυτό το κεφάλαιο αναλύεται ο τρόπος απεικόνισης της κίνησης που καταγράφουν οι συσκευές που δημιουργήσαμε, αφού προηγουμένως γίνει μία εισαγωγή στη 3D μοντελοποίηση και σχεδιοκίνηση καθώς και στην περιγραφή των βασικών λειτουργιών του λογισμικού που χρησιμοποιήθηκε στην εργασία.

4.1 Μοντελοποίηση 3D

Η μοντελοποίηση 3D είναι η διαδικασία, κατά την οποία αναπτύσσεται μια μαθηματική αναπαράσταση κάθε τρισδιάστατης επιφάνειας άψυχων ή έμψυχων αντικειμένων, μέσω εξειδικευμένου λογισμικού, παράγοντας ένα 3D μοντέλο. Τα 3D μοντέλα αντιπροσωπεύουν ένα επίσης 3D αντικείμενο χρησιμοποιώντας μια συλλογή σημείων και άλλων πληροφοριών, στο τρισδιάστατο χώρο, τα οποία συνδέονται μεταξύ τους με διάφορες γεωμετρικές οντότητες όπως τρίγωνα, ευθύγραμμα τμήματα, καμπύλες, κλπ. Τα μοντέλα μπορούν να δημιουργηθούν είτε χειροκίνητα είτε με αλγοριθμικές διαδικασίες (procedural modeling) ή μέσω σάρωσης (model scanning).

Στην παρούσα εργασία δεν θα χρησιμοποιήσουμε τη μοντελοποίηση, εφόσον θα εφαρμόσουμε την απεικόνιση της κίνησης που καταγράφουν οι συσκευές που δημιουργήσαμε, σε ένα έτοιμο μοντέλο 3D.

4.2 Σχεδιοκίνηση τριών διαστάσεων (3D animation)

Ως animation (στα ελληνικά αποδίδεται με τον όρο σχεδιοκίνηση), μπορούμε να ορίσουμε την απόδοση κίνησης σε ένα τρισδιάστατο μοντέλο (3D model). Η δημιουργία ενός 3D animation περιλαμβάνει 3 βασικά στάδια:

- Μοντελοποίηση
- Απόδοση σχεδιοκίνησης
- Απόδοση γραφικών

Στο πρώτο στάδιο λοιπόν, πραγματοποιείται η *μοντελοποίηση*, η οποία όπως προαναφέραμε, παράγει ένα μοντέλο τριών διαστάσεων (3D model). Υπάρχουν αρκετά λογισμικά με τα οποία μπορούμε να δημιουργήσουμε τέτοια μοντέλα (πχ. Photoshop, 3ds Max, Maya, Blender, 3D Slash κλπ.). Επίσης υπάρχει η δυνατότητα να βρούμε έτοιμα τέτοια μοντέλα στο διαδίκτυο. Πρόκειται για μία περίπλοκη διαδικασία καθώς με τη δημιουργία κάθε αντικειμένου, θα πρέπει να υπολογίζουμε την αλληλεξάρτησή του με τα υπόλοιπα αντικείμενα, καθώς και την προτεραιότητα κάθε αλληλεπίδρασης.

Στο δεύτερο στάδιο, που είναι και η ουσία του animation, έχουμε την *απόδοση σχεδιοκίνησης*. Σε αυτό το στάδιο εμφανίζεται η διαδραστικότητα του 3D μοντέλου και γίνεται η προσπάθεια της όσο το δυνατόν ρεαλιστικότερης κίνησης του. Επομένως, εδώ έχουμε την απόδοση ψυχής (anima), ζωντανίας, κίνησης σε γραφικά στατικά αντικείμενα, όπως είναι τα 3D μοντέλα. Τα αντικείμενα ή τμήματα των αντικειμένων μετακινούνται στον τρισδιάστατο χώρο, με τέτοιο τρόπο ώστε να προσδίδεται μια ρεαλιστικότητα στην κίνησή τους.

Στο τρίτο στάδιο έχουμε την *απόδοση γραφικών* (rendering). Πρόκειται για τη διαδικασία ρεαλιστικής απόδοσης των μοντέλων, των περιβαλλόντων στα οποία βρίσκονται, με τη προσθήκη χρωμάτων, υφών, φωτισμού και σκιάσεων. Με αυτό τον τρόπο τα άψυχα αντικείμενα παίρνουν μια πιο ρεαλιστική μορφή, η οποία είναι κοντά στην μορφή που έχουν στον πραγματικό κόσμο. Το πρόγραμμα που χρησιμοποιείται για τη διαδικασία αυτή ονομάζεται renderer και περιέχεται σε όλα τα λογισμικά δημιουργίας 3D animation.

Τέλος, θα πρέπει να επισημάνουμε ότι η δημιουργία της κίνησης είναι μία ψευδαίσθηση που δημιουργείται με την αλληπάλληλη προβολή με μεγάλη ταχύτητα, διαδοχικών στατικών εικόνων (δισδιάστατων ή τρισδιάστατων καρέ-πλαισίων) που διαφέρουν ελάχιστα μεταξύ τους. Ο λόγος που χρησιμοποιείται αυτή η τεχνική είναι ότι στη φυσιολογία του ανθρώπινου ματιού υπάρχει το λεγόμενο μετείκασμα, σύμφωνα με το οποίο, μία εικόνα που βλέπουμε παραμένει, μετά την παρατήρηση της στον αμφιβληστροειδή χιτώνα για ένα πολύ μικρό χρονικό διάστημα (1/12 του δευτερολέπτου).

4.3 Λογισμικό Unity

Σε αυτό το υποκεφάλαιο περιγράφεται το λογισμικό το οποίο χρησιμοποιήσαμε για την εργασία και το λόγο για τον οποίο το επιλέξαμε.

4.3.1 Εισαγωγή στο Unity

Το Unity είναι μια μηχανή ανάπτυξης παιχνιδιών (game engine). Με άλλα λόγια, πρόκειται για μια πλατφόρμα πολλαπλών χρήσεων (cross-platform), με την οποία μπορούμε να δημιουργήσουμε παιχνίδια (video games) και προσομοιώσεις (simulations) για υπολογιστές, κονσόλες (consoles) και κινητά τηλέφωνα (smartphones). Η υλοποίηση αυτής της πλατφόρμας πραγματοποιήθηκε από την εταιρεία Unity Technologies και υποστηρίζει γραφική αναπαράσταση δύο (2D) και τριών (3D) διαστάσεων. Το Unity υποστηρίζει τη δημιουργία project για 27 διαφορετικών ειδών πλατφόρμες (iOS, android, Windows, MAC, Linux, PlayStation 4, Xbox One, Android TV κλπ.).



Εικόνα 4.1: Logo λογισμικού Unity

Ο λόγος που επέλεξα το συγκεκριμένο λογισμικό, το οποίο ειδικεύεται στην ανάπτυξη video games, είναι ότι πρόκειται για ένα περιβάλλον το οποίο προσφέρει μεγάλη γκάμα προβολής και επεξεργασίας μοντέλων τριών διαστάσεων (3D) σε animation. Επομένως, θα μπορούσε η κίνηση του χεριού να απεικονίζεται σε ένα τρισδιάστατο μοντέλο χεριού και με τον κατάλληλο κώδικα σε C# να πραγματοποιηθεί μία ρεαλιστική αναπαράσταση της κίνησης σε πραγματικό χρόνο (real time).

4.3.2 Πλεονεκτήματα Unity

Η επιλογή της συγκεκριμένης πλατφόρμας έγινε διότι προσφέρει μια σειρά από σημαντικά πλεονεκτήματα. Το κυριότερο πλεονέκτημα αυτής της ολοκληρωμένης πλατφόρμας, είναι ότι η βασική του έκδοση (για προσωπική χρήση) διατίθεται δωρεάν στο διαδίκτυο, στη σελίδα της εταιρείας (<https://unity3d.com/>). Άλλο σημαντικό πλεονέκτημα είναι ότι αυτή η πλατφόρμα διαθέτει ένα διαδικτυακό τόπο καταστήματος (asset store), όπου μπορεί να βρει κανείς είτε με πληρωμή είτε εντελώς δωρεάν έτοιμα αντικείμενα, 3D μοντέλα, κώδικες ακόμα και ολόκληρα πακέτα (assets) για χρήση και επεξεργασία, χρήσιμα για το project που θέλει να υλοποιήσει. Τέλος πρόκειται για ένα ολοκληρωμένο περιβάλλον, στο οποίο ο χρήστης έχει τη δυνατότητα με drag and drop να δημιουργεί νέα αντικείμενα, να προσδίδει ιδιότητες σε αυτά, να αλλάζει τις ρυθμίσεις που έχουν ή να τα παραμετροποιεί προγραμματιστικά γράφοντας script σε κώδικα C#.

4.3.3 Έτοιμα μοντέλα 3D / Asset store

Το Unity Asset store είναι ο διαδικτυακός τόπος στον οποίο η εταιρεία του συγκεκριμένου λογισμικού παρέχει πολλά διαφορετικά assets. Τα asset αυτά δημιουργούνται από άλλους προγραμματιστές (developers) και προσφέρονται είτε σε κάποιο κόστος είτε δωρεάν, ανάλογα με την πολυπλοκότητα και τον προγραμματιστή ο οποίος το δημιούργησε. Τα asset αυτά μπορεί να είναι τρισδιάστατα μοντέλα (3D models) ή ολοκληρωμένα πακέτα χαρακτήρων που να περιλαμβάνουν εκτός από το 3D μοντέλο, χρώμα (color), υλικό (material), παραμέτρους, ακόμα και κώδικα (scripts) που να συνοδεύει τμήματα του asset. Εντός του Unity Editor, υπάρχει παράθυρο που συνδέεται με το Asset Store και προσφέρει τη δυνατότητα εισαγωγής των asset που έχουμε κατεβάσει. Το μόνο που έχουμε να κάνουμε είναι import το asset που έχουμε στα αποθηκευμένα κι από εκεί και πέρα μπορούμε να το διαχειριστούμε όπως θέλουμε. Επομένως για να αναζητήσουμε κάποιο asset που θέλουμε να κατεβάσουμε δωρεάν ή να αγοράσουμε, μπορούμε να το κάνουμε είτε κατευθείαν μέσω του editor του λογισμικού, είτε μέσω της ιστοσελίδας: <https://assetstore.unity.com/>.

4.3.4 Βασικές έννοιες και λειτουργίες περιβάλλοντος Unity

Ο τρόπος υλοποίησης της απεικόνισης της κίνησης του χεριού στο λογισμικό του Unity, μπορεί να γίνει πιο κατανοητός, αν εξηγήσουμε κάποιες βασικές έννοιες και λειτουργίες που υπάρχουν στο περιβάλλον του.

Οι 3 βασικότερες έννοιες στο περιβάλλον αυτό είναι οι εξής:

- Models
- Materials
- Prefabs

Τα *Models* είναι τρισδιάστατα ή δισδιάστατα αντικείμενα, τα οποία χρησιμοποιούμε για τη δημιουργία της σκηνής (scene) που θέλουμε να αναπαραστήσουμε.

Τα *Materials* είναι assets, τα οποία ορίζουν τον τρόπο με τον οποίο θα εμφανίζεται ένα αντικείμενο. Σε αυτό συνδέονται τα textures (που θα δούμε παρακάτω) και ρυθμίζονται σχετικές ιδιότητες.

Τα *Prefabs* είναι ένα σύνολο από αντικείμενα και components που αποθηκεύονται σαν ένα αντικείμενο στον φάκελο των assets του project μας. Στόχος τους είναι η εύκολη δημιουργία πολύπλοκων αντικειμένων πολλαπλές φορές, χωρίς να χρειάζεται να επαναπροσδιορίζονται τα components και οι ρυθμίσεις τους.

Άλλες έννοιες για κατανόηση του περιβάλλοντος είναι:

- Textures
- Sprites
- Animations
- Animator
- User Interface (UI)

Τα *Textures* είναι εικόνες που χρησιμοποιούνται για να «ντύσουν» ένα τρισδιάστατο αντικείμενο. Υπάρχουν πολλοί τύποι texture (diffuse, normal map, height map etc).

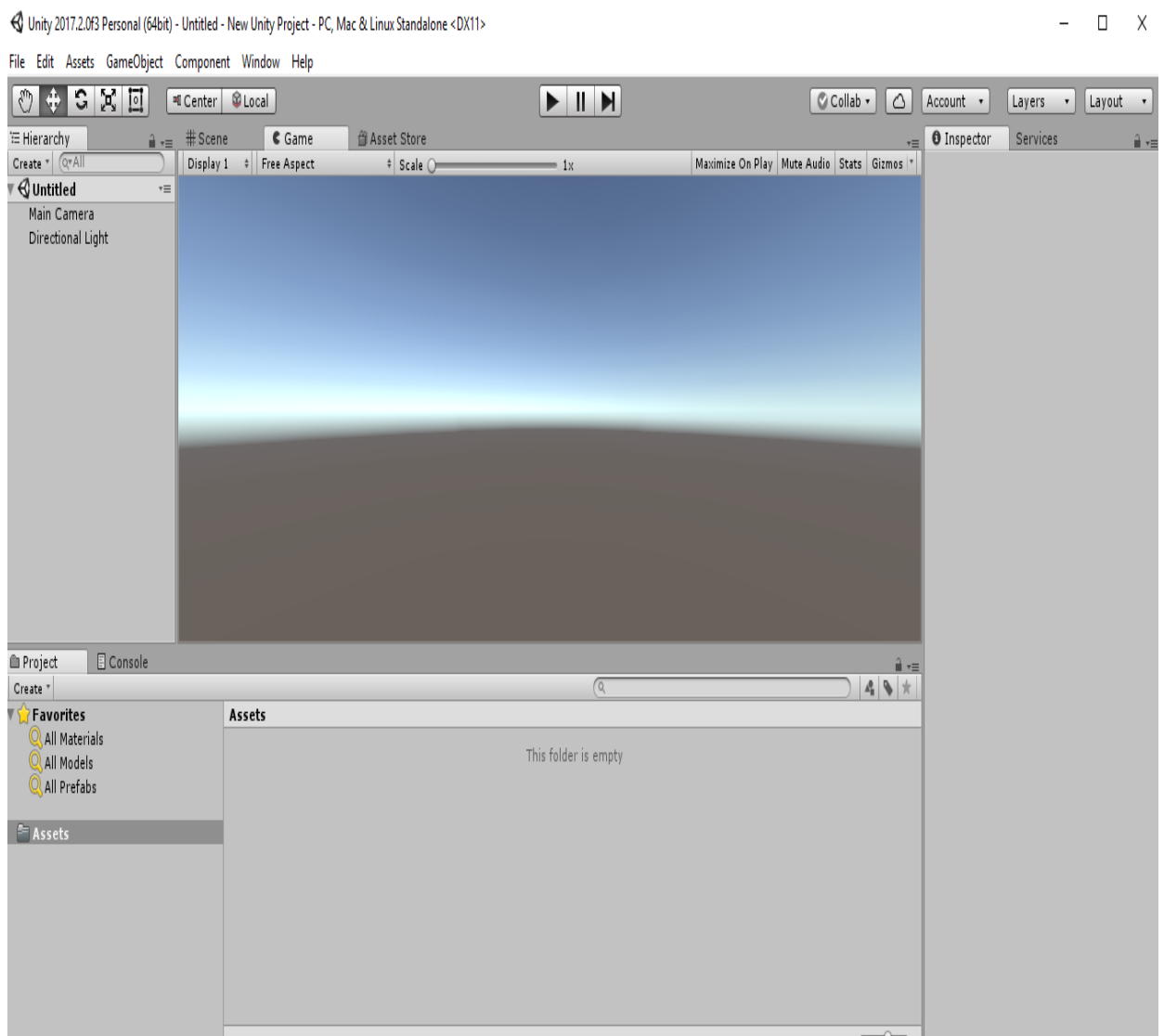
Τα *Sprites* είναι εικόνες που χρησιμοποιούνται ως αναπαραστάσεις αντικειμένων σε δισδιάστατα περιβάλλοντα (2D).

Ασύρματη καταγραφή κίνησης χεριού

Τα *Animations* είναι μία αλληλουχία από διαφορετικά frames, με το κάθε frame να ορίζει μια διαφορετική κατάσταση του αντικειμένου, με αποτέλεσμα τη ψευδαίσθηση της κίνησης και της αλλαγής. Όσο περισσότερα frames έχει ένα animation, τόσο πιο λεπτομερές και ρεαλιστικό θα ναι το αποτέλεσμα.

Ο *Animator* είναι component που διαχειρίζεται όλα τα animations ενός αντικειμένου. Υπάρχει ειδικό παράθυρο animator για τη διαχείριση του animation.

User Interface (UI) είναι γραφικά στοιχεία που βρίσκονται στον editor του Unity, όπως καμβάς (canvas) κείμενα (texts), κουμπιά (buttons), μενού (menu), εικόνες (pictures) κλπ.



Εικόνα 4.2: Παράθυρα (windows) περιβάλλοντος Unity

Παρατηρώντας την παραπάνω εικόνα, βλέπουμε ότι το περιβάλλον του Unity είναι χωρισμένο σε διάφορα παράθυρα (windows).

Ξεκινώντας, έχουμε το παράθυρο *Hierarchy*. Σε αυτό βρίσκονται όλα τα αντικείμενα (objects) τα οποία υπάρχουν στη σκηνή μας. Τα 2 αρχικά προκαθορισμένα (default) αντικείμενα που υπάρχουν σε αυτό το παράθυρο, πριν προσθέσουμε οτιδήποτε άλλο, είναι η κεντρική κάμερα (Main Camera), η οποία λειτουργεί σαν μάτια του χρήστη όταν το project ξεκινήσει να τρέχει (play) και το κατευθυντικό φως (Directional Light) το οποίο δημιουργεί τις σκιάσεις που χρειαζόμαστε για ρεαλιστικότητα.

Πιο κάτω βρίσκεται το παράθυρο *Project*, το οποίο έχει έναν φάκελο που ονομάζεται *Assets*. Εκεί βρίσκονται όλα τα *Assets* (objects, materials, textures scripts), τα οποία θέλουμε να χρησιμοποιήσουμε στο project μας. Μπορούμε να κάνουμε εισαγωγή (import) κατεβασμένων assets από το Asset Store ή χειροκίνητα να βρούμε το φάκελο asset του συγκεκριμένου project στον υπολογιστή μας και να βάλουμε μέσα ότι asset χρειαζόμαστε. Όλα τα assets που θα χρειαστούμε πρέπει να βρίσκονται μέσα σε αυτόν τον φάκελο. Από εκεί κι έπειτα για να τα χρησιμοποιήσουμε μπορούμε απλά να τα σύρουμε (drag and drop) στο ακριβώς από πάνω παράθυρο *Hierarchy*.

Δίπλα από το *Project* υπάρχει η καρτέλα *Console*, την οποία μπορούμε να σύρουμε σε άλλο σημείο του περιβάλλοντος και να εμφανίζεται ως ξεχωριστό παράθυρο (το ίδιο ισχύει με όλες τις υπόλοιπες καρτέλες). Το συγκεκριμένο παράθυρο είναι η κονσόλα που μας εμφανίζει τα αποτελέσματα (output) της εκτέλεσης του project μας. Σε αυτό το παράθυρο εμφανίζονται και τα σφάλματα (errors) τα οποία οδηγούν σε μη ομαλή λειτουργία του project.

Κεντρικά βρίσκεται το παράθυρο *Scene* (σκηνή). Αυτό είναι το κεντρικό παράθυρο στο οποίο βρίσκονται όλα τα αντικείμενα που τοποθετούμε στο παράθυρο *Hierarchy*. Εκεί μπορούμε να σύρουμε τα αντικείμενα στο σημείο που θέλουμε, με τη σειρά και φορά την οποία θέλουμε. Μπορούμε επίσης να τους αλλάξουμε σχήμα και μέγεθος. Είναι με άλλα λόγια ο τρισδιάστατος χώρος, τον οποίο μπορούμε να περιεργαστούμε από διαφορετικές οπτικές γωνίες για το επιθυμητό αποτέλεσμα. Σε αυτόν τον χώρο υπάρχουν εξ' ορισμού 2 αντικείμενα: η κάμερα και το φως σκίασης που όπως είπαμε καθορίζουν τον τρόπο προβολής του τελικού αποτελέσματος.

Το τελικό αποτέλεσμα του project μας εμφανίζεται στο παράθυρο *Game*, όταν ο χρήστης τρέξει (run/play) το project. Πριν τρέξει το πρόγραμμα στο παράθυρο αυτό δεν υπάρχει καμία διαδραστικότητα, καθώς απλά εμφανίζεται η οπτική της κεντρικής κάμερας (display), που βρίσκεται στο παράθυρο *Scene*.

Η καρτέλα *Asset Store* είναι αυτή στην οποία βρίσκουμε τα κατεβασμένα asset μας ή αναζητούμε καινούργια, για να τα προσθέσουμε στο project μας.

Το παράθυρο *Inspector*, που βρίσκεται συνήθως δεξιά, είναι αυτό που μας επιτρέπει να πραγματοποιήσουμε διαφόρων ειδών ρυθμίσεις. Όπως δηλώνει και το όνομά του είναι η επιθεώρηση κάθε αντικειμένου. Όταν επιλέγουμε ένα αντικείμενο σε αυτό το παράθυρο εμφανίζονται: οι συναρτήσεις του, οι διαστάσεις του, η περιστροφή, η θέση του, τα υλικά και χρώματα που ενδεχομένως του έχει προσδώσει ο δημιουργός, ακόμα και κώδικες (script) με τους οποίους αποκτά διαδραστικότητα και επιπλέον ιδιότητες. Έτσι σε αυτό το παράθυρο μπορούμε να αποδώσουμε όλα τα χαρακτηριστικά που θέλουμε στο αντικείμενό μας.

Τέλος πάνω από αυτά τα παράθυρα υπάρχουν ορισμένα κουμπιά εργαλείων (tools), με τα οποία μπορούμε να διαχειριστούμε τα αντικείμενά μας μέσα στη σκηνή (*Scene*) και τα κουμπιά έναρξης (play) και παύσης (pause) του project.

4.4 Υλοποίηση απεικόνισης

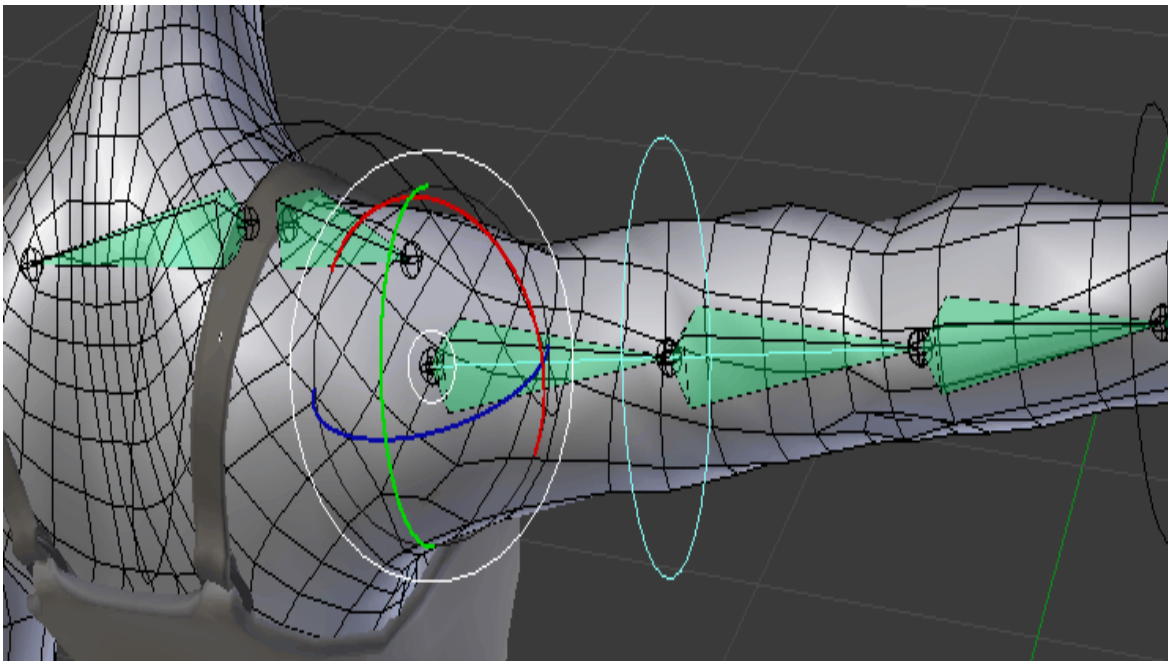
Σε αυτό το υποκεφάλαιο περιγράφεται ο τρόπος με τον οποίο πραγματοποιήθηκε η αναπαράσταση της κίνησης, η οποία καταγράφεται από τις συσκευές που δημιουργήθηκαν.

4.4.1 Επιλογή 3d μοντέλου

Για να πραγματοποιηθεί η αναπαράσταση της κίνησης του χεριού (από τον ώμο μέχρι τον καρπό) που καταγράφουν οι 2 συσκευές που δημιούργησα, θα έπρεπε να βρω ένα κατάλληλο τρισδιάστατο μοντέλο (3D). Επομένως θα έπρεπε να βρω ένα rigged arm pack (πακέτο μοντέλου χεριού) που να αποτελεί ένα ολοκληρωμένο μοντέλο ενός χεριού από τον ώμο έως τα άκρα για την απόδοση κίνησης. Τι σημαίνει όμως rigged;

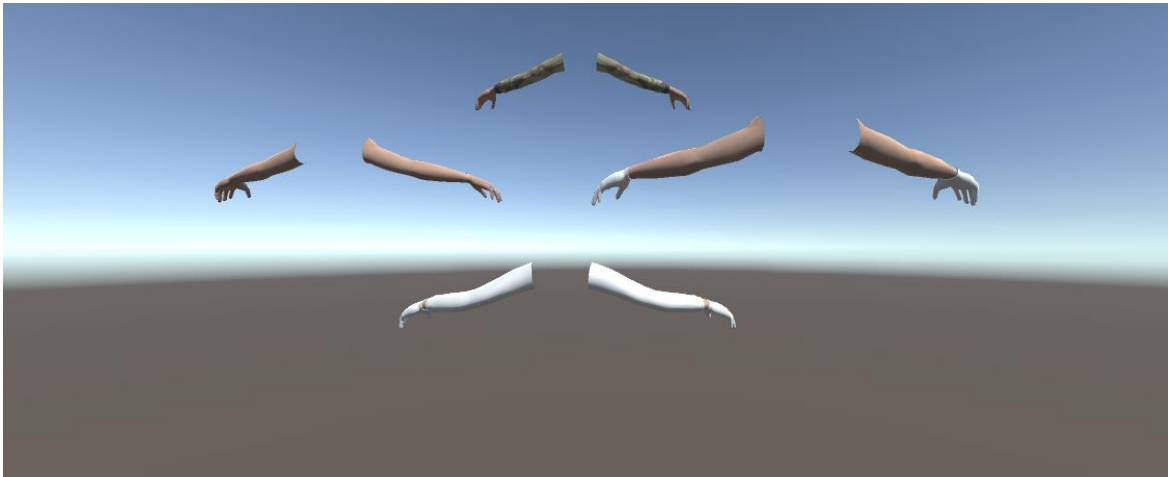
Γενικότερα, στον τομέα της μοντελοποίησης, rigging είναι η διαδικασία κατά την οποία ο χαρακτήρας προετοιμάζεται για την απόδοση κίνησης. Δημιουργείται λοιπόν μια ιεραρχική σκελετική δομή, καθώς και ένα σύνολο κανόνων και προτεραιοτήτων που χαρακτηρίζουν την ανθρώπινη μηχανική του σώματος. Στην απόδοση της κίνησης η σωστή δημιουργία του rig του χαρακτήρα αποτελεί μια απαραίτητη διαδικασία, ώστε ο χαρακτήρας να μπορεί να προσομοιώσει την ανθρώπινη κίνηση.

Συγκεκριμένα, όσον αφορά το χέρι, το μοντέλο πρέπει να ανταποκρίνεται στις προϋποθέσεις που ορίζει η κίνηση που πρόκειται να αποδοθεί. Για παράδειγμα οι αρθρώσεις θα πρέπει να κινούνται όλες στη σωστή κατεύθυνση καθώς και να διέπονται από μία ιεραρχία, που διέπει την ανθρώπινη φυσιολογία. Επίσης στις κλειδώσεις θα πρέπει να υπάρχει και αλλαγή της προσομοίωσης του δέρματος (εφόσον δεν μιλάμε για σκελετό), για παράδειγμα στον αγκώνα το δέρμα στην εξωτερική πλευρά της κλείδωσης θα διασταλεί, ενώ στην εσωτερική θα συσταλεί. Επομένως ο δημιουργός του μοντέλου θα πρέπει να γνωρίζει την ανατομία του χαρακτήρα, ώστε το σχήμα και η πλέξη των πολυγώνων να διαμορφωθούν έτσι ώστε να προσδίδουν ελαστικότητα στα μέρη που πρόκειται να παραμορφωθούν κατά την κίνηση.



Εικόνα 4.3: Διαδικασία rigging τρισδιάστατου μοντέλου χεριού

Το μοντέλο λοιπόν που επέλεξα, είναι ένα δωρεάν πακέτο τρισδιάστατου μοντέλου χεριού, το οποίο διανέμεται ελεύθερα για προσωπική ή εμπορική χρήση στην διεύθυνση: <https://www.cgtrader.com/items/696175/download-page>. Ο τίτλος του αρχείου είναι Fps Arm Pack και περιέχει rigged 3D objects δεξιού και αριστερού χεριού, με μανίκι ή χωρίς και με γάντι ή χωρίς, καθώς και τα αντίστοιχα textures.



Εικόνα 4.4: Objects πακέτου Fps Arm Pack

Το object, στο οποίο επέλεξα να υλοποιήσω την αναπαράσταση, είναι αυτό του δεξιού γυμνού χεριού με το γάντι, καθώς η καταγραφή θα πραγματοποιηθεί στα σημεία από τον ώμο έως τον καρπό κι έτσι θα είναι πιο ευδιάκριτο ότι δεν υπάρχει καταγραφή στο σημείο κάτω από τον καρπό (που καλύπτεται με γάντι). Το συγκεκριμένο object βρίσκεται στο φάκελο Fps Arms Pack \ Arms_PayDay_Style \ No_Sleeve με την ονομασία ArmsPayDayHigh.

4.4.2 Ανάλυση επιμέρους κομματιών 3d μοντέλου

Εφόσον εξασφαλίσουμε ότι το μοντέλο είναι rigged, ανταποκρίνεται δηλαδή σωστά στην κίνηση του χεριού, εξασφαλίζουμε και ότι θα περιέχει τις αρθρώσεις (joints) που χρειαζόμαστε. Οι 2 αρθρώσεις που χρειαζόμαστε σε αυτή την εργασία και λειτουργούν ως διανύσματα (vectors) στο project μας, είναι οι εξής:

- Αυτή που ενώνει τις κλειδώσεις ώμου και αγκώνα
- Αυτή που ενώνει τις κλειδώσεις αγκώνα και καρπού

Λόγω του ότι το μοντέλο που επέλεξα είναι ένα rigged arm pack, δηλαδή πακέτο αρθρώσεων χεριού που ανταποκρίνονται στην φυσική κίνηση του χεριού, δεν χρειάστηκε από πλευράς κώδικα να γίνει περαιτέρω ανάλυση της κινηματικής του χεριού. Η κάθε άρθρωση, ως ξεχωριστό object με την ιεραρχική δομή του ανθρώπινου σώματος, διαθέτει τη δικιά της συνάρτηση transform, η οποία διαθέτει την παράμετρο rotation στους 3 άξονες (x, y, z).

Αρκούσε λοιπόν, για την σωστή καταγραφή, οι μετρήσεις της μίας συσκευής που έφτιαξα να μεταβάλλουν τις τιμές rotation της πάνω άρθρωσης και οι μετρήσεις της άλλης συσκευής να μεταβάλλουν τις τιμές rotation της κάτω άρθρωσης. Όπως αναφέραμε στο κεφάλαιο 3.4.2, είχαμε ρυθμίσει η μία bluetooth συσκευή να στέλνει μετρήσεις στη σειριακή πόρτα COM5 και η άλλη στην COM8. Αυτή που επικοινωνούσε με την COM5 την ονομάσαμε up, διότι θα είναι αυτή που θα τοποθετηθεί στην πάνω μεριά του χεριού (ανάμεσα σε ώμο-αγκώνα), ενώ την άλλη down γιατί θα τοποθετηθεί στην κάτω μεριά του χεριού (ανάμεσα σε αγκώνα-καρπό).

Όσον αφορά το μοντέλο μας παρατηρούμε αποτελείται από διάφορα τμήματα του χεριού. Τα 2 τμήματα του μοντέλου που μας ενδιαφέρουν είναι το RightArm για την καταγραφή της κίνησης της πάνω άρθρωσης και το RightForeArm_ για την καταγραφή της κίνησης της κάτω άρθρωσης. Κάθε τμήμα του μπορεί να επιλεγθεί και χαρακτηρίζεται από τη συνάρτηση transform. Σε αυτή τη συνάρτηση υπάρχει η παράμετρος rotation μας ενδιαφέρει.

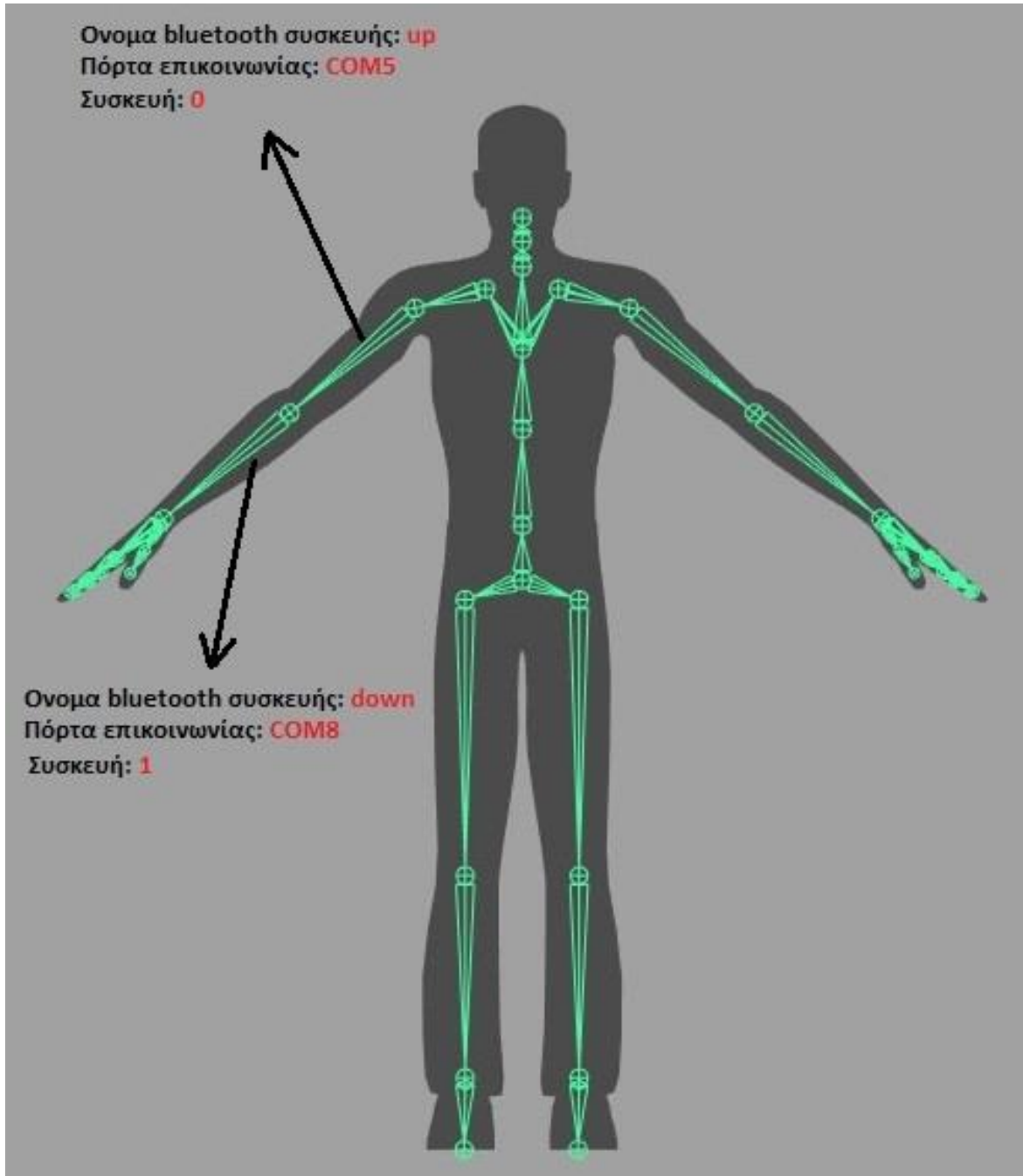
Επομένως η σωστή απόδοση της κίνησης θα πραγματοποιηθεί, εάν:

- Οι τιμές που καταγράφει ο αισθητήρας της πάνω συσκευής (άρα μετρήσεις της σειριακής πόρτας COM5) μεταβάλλουν το rotation του RightArm
- Οι τιμές που καταγράφει ο αισθητήρας της κάτω συσκευής (άρα μετρήσεις της σειριακής πόρτας COM8) μεταβάλλουν το rotation του RightForeArm_

Αυτό μπορεί εύκολα να πραγματοποιηθεί με τη χρήση 2 script C# (ένα για το RightArm και ένα για το RightForeArm_). Υπάρχουν κατάλληλες εντολές, οι οποίες διαβάζουν τις τιμές που εισέρχονται στις σειριακές πόρτες και κατάλληλες συναρτήσεις που αναγνωρίζουν αυτές τις τιμές ως τετραδόνια (quaternions) ή γωνίες Euler και τις προσδίδουν αντίστοιχα στις τιμές περιστροφών των αξόνων

Ασύρματη καταγραφή κίνησης χεριού

κάθε τμήματος (rotation.x, rotation.y, rotation.z), με αποτέλεσμα το εκάστοτε τμήμα να περιστρέφεται ρεαλιστικά με τον τρόπο που καταγράφουν οι συσκευές μας σε πραγματικό χρόνο (real time).



Εικόνα 4.5: Σημεία αρθρώσεων καταγραφής κίνησης

4.4.3 Περιστροφή (Rotation) μοντέλου στο χώρο

Στο Unity από πλευράς κώδικα δημιουργήσα 2 πανομοιότυπα script (το μόνο που αλλάζει είναι η πόρτα που ανοίγουν) γραμμένα σε C#.

Το ένα ονομάζεται *up.c*, τοποθετείται στο Right Arm (με drag and drop) και αυτό που κάνει είναι να ανοίγει την ασύρματη σειριακή πόρτα COM5, να διαβάζει τα quaternions που έρχονται από τον αισθητήρα της συσκευής (0), να τα μετασχηματίζει στο σύστημα συντεταγμένων του Unity και να μεταβάλλει τα rotation (x, y, z) του Right Arm. Έτσι το Right Arm περιστρέφεται ανάλογα με την περιστροφή του πρώτου αισθητήρα.

Το άλλο ονομάζεται *down.c*, τοποθετείται στο RightForeArm_ (επίσης με drag and drop) και αυτό που κάνει είναι να ανοίγει την ασύρματη σειριακή πόρτα COM8, να διαβάζει τα quaternions που έρχονται από τον αισθητήρα της συσκευής (0), να τα μετασχηματίζει στο σύστημα συντεταγμένων του Unity και να μεταβάλλει τα rotation (x, y, z) του RightForeArm_. Έτσι το και το RightForeArm_ περιστρέφεται ανάλογα με την περιστροφή του δεύτερου αισθητήρα.

Επομένως συνολικά τα 2 script, όταν το project τρέχει (run) υλοποιούν σε μεγάλο βαθμό ρεαλιστικότητας την απεικόνιση της κίνησης. Εάν το Animation Type του μοντέλου μας είναι Humanoid, τότε η κίνηση είναι πιο ρεαλιστική καθώς τηρούνται όλες οι ιεραρχίες της φυσιολογίας του ανθρώπινου σώματος, κάτι το οποίο δεν ισχύει για το μοντέλο της εργασίας αυτής.

4.4.4 Επεξήγηση κώδικα απεικόνισης

Η εντολή *Quaternion.Identity* επιστρέφει τα quaternions του αντικειμένου σύμφωνα με το παγκόσμιο πλαίσιο αναφοράς και χαρακτηρίζεται ως no rotation, δηλαδή δεν περιστρέφει απλά ευθυγραμμίζει. Χρησιμοποιείται για calibration, δηλαδή βαθμονόμηση για να έχουν νόημα οι μετρήσεις μας. Αυτά τα quaternions τα αποθήκευσα στην μεταβλητή *rotationOffset*, επομένως έχουμε:

```
//used for calibration  
Quaternion rotationOffset = Quaternion.identity;
```

Τα quaternions που παίρνω από τον αισθητήρα τα αποθηκεύω στην μεταβλητή q . Τα 4 πρώτα quaternions είναι ίδια με αυτά του rotationOffset, από εκεί κι έπειτα όμως οι τιμές του q μεταβάλλονται όσο λαμβάνω τιμές από τις συσκευές.

Αφού αρχικοποιήσω σειριακή πόρτα (COM5 για το up και COM8 για το down) και ορίσω όλες τις παραμέτρους της (name, buffer size, new line characters, read timeout), την ανοίγω (αν δεν ανοίξει εμφανίζω μήνυμα σφάλματος). Στη συνέχεια στέλνω ένα χαρακτήρα για να ενεργοποιηθεί ο αισθητήρας μου, καθώς αν δεν σταλεί δεν θα ξεκινήσει να στέλνει τιμές.

Εφόσον ο αισθητήρας στέλνει τιμές, με μία συνθήκη if εφαρμόζεται βαθμονόμηση, ώστε να ξεκινήσουν οι τιμές να περιστρέφουν το τρισδιάστατο μοντέλο. Η βαθμονόμηση ξεκινάει όταν πατηθεί το πλήκτρο 'C' και πραγματοποιείται πολλαπλασιάζοντας τα αρχικά quaternions με τα αντίστροφά τους. Έτσι έχουμε:

```
//If we press it begins calibration and rotation
if (Input.GetKeyDown(KeyCode.C) && calibrated == false)
{
    rotationOffset = transform.rotation * Quaternion.Inverse(q);
    calibrated = true;
}
```

Κάθε πακέτο μετρήσεων (4 τιμές quaternion) που έρχεται από τον αισθητήρα, αποθηκεύεται στο string με την ονομασία value. Κάνοντας split αυτό το string με βάση το χαρακτήρα '/', ο οποίος είναι αυτός που διαχωρίζει τις τιμές quaternion, μετατρέπουμε strings που έχουν υποστεί split σε τύπο float και έτσι παίρνουμε τις 4 τιμές που θέλουμε στις μεταβλητές $q.w$, $q.x$, $q.y$, $q.z$:

```
// we split the string in 4 pieces (separated with /t)
// to find 4 measurements of quaternion (q.w, q.x, q.y, q.z)
string value = sp.ReadLine();
string[] vec4 = value.Split('\t');
q.w = float.Parse(vec4[0]);
q.x = float.Parse(vec4[1]);
q.y = float.Parse(vec4[2]);
q.z = float.Parse(vec4[3]);
```

Έπειτα, σειρά έχει ο μετασχηματισμός από το σύστημα συντεταγμένων του αισθητήρα στο σύστημα συντεταγμένων του περιβάλλοντος Unity. Αυτό επιτυγχάνεται με τη χρήση διανυσμάτων 3 αξόνων `Vector3`. Μετατρέπω τα `quaternions` του αισθητήρα σε γωνίες (`angle`) των 3 αξόνων (`axis`) του `rotation` του Unity. Δημιουργώ νέα `quaternions` που είναι μετασχηματισμένα σε γωνίες περιστροφής των αξόνων (`x`, `y`, `z`) του Unity και τα αποθηκεύω στη ίδια μεταβλητή `q`. Έτσι οι νέες τιμές των `quaternion` (`q.w`, `q.x`, `q.y`, `q.z`) θα μπορούν να αλληλεπιδρούν σωστά με το σύστημα περιστροφής του Unity. Έχουμε λοιπόν:

```
//transform sensor's coordinate system into unity's coordinate system
q.ToAngleAxis(out angle, out axis);
q = Quaternion.AngleAxis(angle, new Vector3(-axis[2], axis[1], axis[0]));
```

Αυτές οι `quaternion` τιμές λοιπόν, που είναι προσαρμοσμένες στο πλαίσιο αναφοράς του Unity, σε συνδυασμό με τις `quaternion` τιμές του παγκόσμιου πλαισίου αναφοράς (`rotationOffset`), μας δίνουν την τελική περιστροφή (`rotation`) του αντικειμένου της συνάρτησης `transform`, το οποίο είναι και το ζητούμενο.

```
if (calibrated)
{
    //it rotates in worldspace and not locally
    transform.rotation = rotationOffset * q;
}
```

Τέλος, όταν έχουμε έξοδο του προγράμματος, κλείνουμε τη σειριακή πόρτα:

```
void OnApplicationQuit()
{
    if (sp != null && sp.IsOpen == true)
        sp.Close();
}
```

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

5.1 Σύνοψη πτυχιακής εργασίας

Η συγκεκριμένη πτυχιακή ήταν μια προσπάθεια κατανόησης των νέων τεχνολογιών, τόσο στον χώρο του υλικού (hardware) όσο και του λογισμικού (software). Πρόκειται μία δύσκολη εργασία, η οποία εμβαθύνει στον κόσμο της αδρανειακής κίνησης και στον τρόπο με τον οποίο μπορούμε να μετρήσουμε με αδρανειακούς αισθητήρες, με τη βοήθεια νέων τεχνολογιών (όπως Arduino). Μεγάλη έμφαση, επίσης δίνεται στο πεδίο της τρισδιάστατης μοντελοποίησης και στην ιδέα του συνδυασμού της μοντελοποίησης με μικροελεγκτές και αισθητήρες, οι οποίοι μπορούν να εκτοξεύσουν τις δυνατότητες του animation σε άλλη διάσταση (4^η!!!). Οι δυσκολίες ξεκίνησαν από την προσπάθεια ασύρματης καταγραφής (μέσω bluetooth) και συνεχίστηκαν στην κατανόηση τετραδονίων, αδρανειακών πλαισίων αναφοράς και προσαρμογής των μετρήσεων στο τρισδιάστατο μοντέλο. Έτσι, δαπανήθηκαν πάρα πολλές ώρες για κατανόηση τόσο θεωρητικών εννοιών όσο και πρακτικών στην κατασκευή και στον κώδικα, αλλά με πολύ προσπάθεια και πολύ υπομονή το αποτέλεσμα ήταν αρκετά ικανοποιητικό.

5.2 Προοπτικές

Η συγκεκριμένη πτυχιακή εργασία, λόγω του ότι ασχολείται με πολλά διαφορετικά πεδία, όπως προαναφέραμε, μπορεί να έχει αντίκρισμα σε πολλούς διαφορετικούς τομείς. Η καταγραφή της κίνησης εν γέννη, είναι κάτι που συναντάται σε πολλά πεδία, όπως πχ στην ιατρική. Θα μπορούσε για παράδειγμα η καταγραφή της κίνησης, σε μια έκδοση πιο λεπτομερών μετρήσεων, να μην περιορίζεται απλά στην αναπαράσταση της κίνησης. Θα μπορούσαν οι μετρήσεις της κίνησης του χεριού ενός ασθενή να καταγράφονται σε μία βάση δεδομένων και να πραγματοποιείται γραφική αναπαράσταση συγκρίνοντας με ιστορικό προηγούμενων μετρήσεων ή με ένα πακέτο προκαθορισμένων από το πρόγραμμα κινήσεων. Έτσι, το αποτέλεσμα των γραφικών παραστάσεων θα προβάλλει τη βελτίωση της κίνησης του ασθενή. Εκτός αυτού, μια άλλη επέκταση της συγκεκριμένης εργασίας θα μπορούσε να είναι στον χώρο των mocap studios. Στο

εξωτερικό υπάρχουν motion capture studio, δηλαδή studio που καταγράφεται η κίνηση ολόκληρου του σώματος ενός ηθοποιού και αποδίδεται σε έναν anime χαρακτήρα για την απόδοση ρεαλιστικότερης κίνησης. Αυτές τις καταγραφές τις χρειάζονται αυτοί που δημιουργούν animations είτε για βιντεοπαιχνίδια, είτε για ταινίες και πραγματοποιούνται με κάμερες καταγραφής κίνησης, κάτι το οποίο κοστίζει αρκετά. Θα μπορούσε μία τέτοια καταγραφή σε πιο απλή μορφή να γίνει με περισσότερες συσκευές, όπως αυτές τις εργασίας μας, οι οποίες θα τοποθετούνταν σε ολόκληρο το σώμα, κι εκτός από την περιστροφή θα κατέγραφαν και την κίνηση μέσα στο χώρο με τη χρήση των μετρήσεων του επιταχυνσιόμετρου. Το μόνο σίγουρό είναι ότι όπου υπάρχει φαντασία δεν υπάρχουν όρια .

ΚΕΦΑΛΑΙΟ 6

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΙΣΤΟΣΕΛΙΔΕΣ

- Accelerometers, gyroscopes, MEMS, yaw-pitch-roll. (Stanford University)
<https://stanford.edu/class/ee267/lectures/lecture9.pdf>
- Αδρανειακή πλοήγηση.
http://users.otenet.gr/~elpida_1/Dionisis_Stefanatos/Gyroskopia_Stefanatos_Shmeiwseis.pdf
- Gimbal lock. http://ikee.lib.auth.gr/record/294871/files/ptuxiakh_v2.pdf
- Gyroscope vs Accelerometer. <https://www.livescience.com/40103-accelerometer-vs-gyroscope.html>
- Quaternions.
<http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/>
- Arduino Hardware. <http://arduino.cc/en/Main/Hardware>
- ATmega328p Microcontroller.
<http://www.atmel.com/devices/atmega328.aspx>
- Adafruit Technology, drivers and datasheet.
<https://www.adafruit.com/product/2000>
- MPU6050 details and datasheet.
<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- 3D μοντελοποίηση ανθρώπινης κίνησης.
<http://www.syros.aegean.gr/de/dpsd05007.pdf>
- Motion capture History.
http://www.streetdirectory.com/travel_guide/160036/computers/history_of_motion_capture_for_computer_character_animation.html
- Motion capture systems.
<http://nemertes.lis.upatras.gr/jspui/handle/10889/10413>
- Motion capture. https://www.xsens.com/tags/motion-capture/?qclid=Ci0KCQjw-uzVBRDkARIsALkZAdmCiS1jpxBkxeeHjPbyZUqLrOz0T3f-5qKbfgJIDGVPBSpLqdBcGxgaAkbsEALw_wcB

- Mocap studios. <http://www.mocaplab.com/>
- Κατανόηση περιβάλλοντος Unity.
https://eclass.uoa.gr/modules/document/file.php/DI413/%CE%94%CE%B9%CE%AC%CF%86%CE%BF%CF%81%CE%B1/Unity3D_2.pdf
- Unity 3d. <https://unity3d.com/>
- Unity asset store. <https://www.assetstore.unity3d.com/>
- Unity editor.
<https://eclass.uoa.gr/modules/document/file.php/DI413/%CE%A0%CE%B1%CF%81%CE%BF%CF%85%CF%83%CE%B9%CE%AC%CF%83%CE%B5%CE%B9%CF%82/Unity%20editor.pdf>