

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

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Αυτοκινούμενο Όχημα με Raspberry pi και Line Camera**

**Θεοφάνης Τσατσαρώνης, Φώτιος Φαρμάκης**

**Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής**

**ΑΘΗΝΑ  
Οκτώβριος 2017**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Αυτοκινούμενο Όχημα με Raspberry pi και Line Camera**

**Θεοφάνης Τσατσαρώνης , A.M:40436**  
**Φώτιος Φαρμάκης , A.M:40456**

**Εισηγητής:** Δρ. Ιωάννης Έλληνας, Καθηγητής

**Εξεταστική Επιτροπή:** .....

**Ημερομηνία εξέτασης:** .....



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Οι κάτωθι υπογεγραμμένοι Τσατσαρώνης Θεοφάνης, του Δημητρίου, με αριθμό μητρώου 40436 και Φαρμάκης Φώτιος του Παναγιώτη, με αριθμό μητρώου 40456 φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε 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 απονείμει 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 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε καταρχήν να ευχαριστήσουμε όλους όσους συνέβαλαν με οποιονδήποτε τρόπο, στην επιτυχή ολοκλήρωση αυτής της πτυχιακής εργασίας. Επιπλέον, θα θέλαμε να ευχαριστήσουμε θερμά τον επιβλέποντα Κύριο Ιωάννη Έλληνα, καθηγητή μας και προϊστάμενο του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων Α.Τ.Ε.Ι Πειραιά Τ.Τ για τη βοήθεια, την καθοδήγηση και τις χρήσιμες συμβουλές που μας παρείχε.

Το μεγαλύτερο ευχαριστώ το οφείλουμε στους γονείς μας, οι οποίοι στήριξαν τις σπουδές μας με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωσή μας.





## ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία έχει σκοπό την ανάπτυξη ενός ρομποτικού συστήματος όπου θα ακολουθεί μια μαύρη γραμμή διαγραμμένη στο δάπεδο. Ο σκοπός αυτός πραγματοποιήθηκε με την βοήθεια τριών βασικών στοιχείων:

1. Την χρήση δύο τροχών, για την μπροστινή και την οπίσθια κίνηση
2. Την χρήση κάμερας (που παίζει το ρόλο του ματιού του ρομποτικού συστήματος) για την συνεχή παραγωγή φωτογραφιών, ώστε να μπορεί το σύστημα να ακολουθηθεί την μαύρη γραμμή.
3. Την χρήση της ηλεκτρονικής πλατφόρμας raspberry pi, για την επεξεργασία των δεδομένων που λαμβάνει από την κάμερα και για την κίνηση των τροχών

Η παρούσα πτυχιακή εργασία επιλέχθηκε επειδή μας άρεσε η ιδέα να ασχοληθούμε με την ρομποτική τεχνολογία και για έναν άλλον κύριο λόγο επειδή μέσω αυτής της τεχνολογίας μπορείς να προσφέρεις στην βιομηχανία και στην ανθρωπότητα. Στην αρχή της πτυχιακής εργασίας αναλύονται οι αρχές της ρομποτικής τεχνολογίας και ποιες είναι οι εφαρμογές της, έπειτα αναφερόμαστε στις τεχνολογίες που χρησιμοποιήσαμε για την επίτευξη της παρούσας πτυχιακής εργασίας. Τέλος, περιγράφονται τα βήματα, η εγγραφή και η ανάλυση του κώδικα που ακολουθήσαμε ώστε να έχουμε το πιο ορθό και το πιο εύκολο τρόπο ανάπτυξης αυτής της εργασίας.

**ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ:** Raspberry pi, Ψηφιακή Επεξεργασία Εικόνας, Προγραμματισμός

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Raspberry pi (Πλακέτα), Python(Γλώσσα Προγραμματισμού), OpenCV (Βιβλιοθήκη Επεξεργασίας Εικόνας), Usb Camera, L293D chip (Ολοκληρωμένο H-bridge), Raspbian (Λειτουργικό)

## ABSTRACT

The diploma thesis aims to develop a robotic system followed by a black line written on the floor. This was done with the help of three key elements

1. The use of two wheels for the front and rear movement
2. The use of a camera (which plays the role of the robotic eye) for the continuous production of photographs, so that the system can follow the black line.
3. The use of the raspberry pi electronic platform to process the data it receives from the camera and to drive the wheels

This diploma thesis was chosen because we liked the idea of dealing with robotic technology and for another reason mainly because through this technology you can offer to industry and humanity. At the beginning of the diploma thesis we analyze the principles of robotic technology and what its applications are, then we refer to the technologies we used to achieve this diplomatic work.

Finally, we describe the steps, the registration and the analysis of the code that we have followed in order to have the most correct and easiest way of developing this work.

**SCIENTIFIC AREA:** Raspberry pi, Digital Image Processing, Programming

**KEYWORDS:** Raspberry pi, Python (Programming Language), OpenCV (Image Processing Library), Usb Camera, L293D chip, Raspbian (Operating System)

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΕΥΧΑΡΙΣΤΙΕΣ</b>	<b>6</b>
<b>ΠΕΡΙΛΗΨΗ</b>	<b>8</b>
<b>ABSTRACT</b>	<b>9</b>
<b>ΚΕΦΑΛΑΙΟ 1</b>	<b>15</b>
1.1 Ρομπότ και Ρομποτική	15
1.1.1 Εισαγωγή	15
1.1.2 Τι Είναι Ρομπोट ;	16
1.1.3 Κατηγορίες των ρομπότ	19
1.1.4 Παρελθόν,παρόν και μέλλον ρομποτικής	20
<b>ΚΕΦΑΛΑΙΟ 2</b>	<b>27</b>
2.1 Υλικά Κατασκευής	27
2.1.1 Raspberry Pi	27
2.1.2 USB Camera	35
2.1.3 Micro SD Card	35
2.1.4 Σάσι (Chassis) με ηλεκτρικούς κινητήρες	36
2.1.5 MPU6050 (Γυροσκόπιο , Επιταχυνσιόμετρο)	38
2.1.6 Το ολοκληρωμένο L298N	43
2.1.7 Breadboard	45
2.1.8 Θηλυκά και αρσενικά καλώδια (jumpers)	45
2.1.9 Εξωτερικός φορτιστής μπαταριών (powerbank) και 9 Volt Μπαταρία	46
2.1.10 Αντάπτορας Wi-fi	46
<b>ΚΕΦΑΛΑΙΟ 3</b>	<b>47</b>
3.1 Ψηφιακή Επεξεργασία Εικόνας	47
3.1.1 Εισαγωγή	47
3.1.2 Ιστορικά Στοιχεία	47
3.1.3 Εφαρμογές	48
3.2 OpenCV και Επεξεργασία Εικόνας	50
3.2.1 Εισαγωγή στην Βιβλιοθήκη OpenCV	50
3.2.2 Δομή και περιεχόμενο της OpenCV	50
3.2.3 Εφαρμογές της OpenCV	51
3.2.4 Βασικές Δομές και Συναρτήσεις της OpenCV	52
3.2.5 Πλεονεκτήματα OpenCV	53
3.3 Η Επεξεργασία Εικόνας στην πράξη	53
3.3.1 Η γλώσσα προγραμματισμού Python στην επεξεργασία εικόνας	53
3.3.2 Φίλτρο Gaussian	54
3.3.3 Χρωματικός χώρος HSV	56
3.3.4 Threshold	58
3.3.5 Εύρεση γραμμής	59

3.3.6 Αναγνώριση σημάτων	61
3.4 Ισορροπία Οχήματος	64
3.4.1 Πως επιτυγχάνεται η ισορροπία	64
3.4.2 Ανάγνωση των δεδομένων από το MPU6050	65
3.4.3 Ο αλγόριθμος PID (Proportional Integral Derivative)	66
3.4.4 Ο διάυλος επικοινωνίας I2C	69
<b>ΚΕΦΑΛΑΙΟ 4</b>	<b>73</b>
4.1 Υλοποίηση - Συναρμολόγηση κατασκευής ,Εγκατάσταση προγραμμάτων.	73
4.1.1 Εγκατάσταση Λειτουργικού Συστήματος στο Raspberry Pi	73
4.1.2 Σύνδεση και πρώτη χρήση του raspberry pi	77
4.1.3 Σύνδεση στο Raspberry Pi μέσω του Wi-fi	82
4.1.4 Εγκατάσταση OpenCV και Python	84
4.1.5 Εγκατάσταση και σύνδεση κάμερας	87
4.1.6 Εγκατάσταση drivers MPU6050 και σύνδεση στο Raspberry pi	89
4.1.7 Σύνδεση του ολοκληρωμένου L298N στο Raspberry pi	92
4.1.8 Στήσιμο Πίστας	94
4.1.9 Συναρμολόγηση Οχήματος	97
<b>ΚΕΦΑΛΑΙΟ 5</b>	<b>103</b>
5.1 Αποτελέσματα, Δυσκολίες που αντιμετωπίσαμε και Βελτιώσεις	103
5.1.1 Αποτελέσματα Εργασίας	103
5.1.2 Δυσκολίες	103
5.1.3 Βελτιώσεις	104
<b>ΚΕΦΑΛΑΙΟ 6</b>	<b>105</b>
6.1 Παράρτημα	105
6.1.1 Ειδικό Λεξιλόγιο και Συντομογραφίες	105
6.1.2 Κώδικας	106
<b>ΚΕΦΑΛΑΙΟ 7</b>	<b>113</b>
7.1 Βιβλιογραφία	113

## Κατάλογος Εικόνων

Εικόνα 1 - Ρομπότ στις Ταινίες	15
Εικόνα 2 - Raspberry Pi	16
Εικόνα 3 - Arduino	16
Εικόνα 4 - Ανεμόμυλος του Edmund Lee	17
Εικόνα 5 - Αισθητήρες	18
Εικόνα 6 - Ρομπότ Ναρκαλιευτής	19
Εικόνα 7 - Drone	19
Εικόνα 8 - Πάπια	21
Εικόνα 9 - Ρομπότ Χελώνα	21
Εικόνα 10 - Stanford Cart	22
Εικόνα 11 - Ρομπότ Sony Dream (SDR)	23
Εικόνα 12 - Τα μέρη του Raspberry Pi	27
Εικόνα 13 - Raspberry Pi Type A	29
Εικόνα 14 - Raspberry Pi Type A+	29
Εικόνα 15 - Raspberry Pi Type B	30
Εικόνα 16 - Raspberry Pi Type B+	31
Εικόνα 17 - Raspberry Pi 2,Raspberry Pi 3,Raspberry Pi Zero	33
Εικόνα 18 - Usb Camera Logitech Webcam C270	35
Εικόνα 19 - Micro SD Sandisk Κάρτα Μνήμης	36
Εικόνα 20 - Σασί	36
Εικόνα 21 - Dc Motors	38
Εικόνα 22 - Περιστρεφόμενος τροχός	39
Εικόνα 23 - 3 Axis Digital Gyro	39
Εικόνα 24 - Επιταχυνσιόμετρο Πίεσης	40
Εικόνα 25 - Πιεζοηλεκτρικό Επιταχυνσιόμετρο	41
Εικόνα 26 - Απλοποιημένο Διάγραμμα Πιεζοηλεκτρικού Επιταχυνσιόμετρου	41
Εικόνα 27 - Επιταχυνσιόμετρο και Γυροσκόπιο (MPU)	42
Εικόνα 28 - Το ολοκληρωμένο L298N	44
Εικόνα 29 - Breadboard (Δοκιμαστική πλακέτα)	45
Εικόνα 30 - Θηλυκά και αρσενικά καλώδια (Jumpers)	45
Εικόνα 31 - Μπαταρία 9V	46
Εικόνα 32 - Powerbank	46
Εικόνα 33 - Wi-fi Αντάπτορας	46
Εικόνα 34 -Πρώτη φωτογραφία με χρήση τεχνικών φωτογραφικής αναπαραγωγής	47
Εικόνα 35 - Φεγγάρι	48
Εικόνα 36 - Lenna Επεξεργασία Εικόνας	48
Εικόνα 37 - Διάφορες μορφές στην επεξεργασία εικόνας	49
Εικόνα 38 - Η δομή και το περιεχόμενο της OpenCV	51

Εικόνα 39 - Αυτόνομο Ρομπωτικό Όχημα από το Πανεπιστήμιο του Στανφορντ	52
Εικόνα 40 -Κατανομή του φίλτρου Gauss σε 2 διαστάσεις	55
Εικόνα 41 - Κατανομή του φίλτρου Gauss σε 3 διαστάσεις	55
Εικόνα 42 - Εικόνα πριν και μετά τη χρήση του φίλτρου Gaussian	56
Εικόνα 43 - Χρωματικός χώρος HSV	57
Εικόνα 44 - Παράδειγμα επεξεργασίας HSV	57
Εικόνα 45 - Τύποι Threshold στο OpenCV	58-59
Εικόνα 46 - Διάγραμμα ελέγχου PID	67
Εικόνα 47 - Αναπαράσταση της ισορροπίας	67
Εικόνα 48 - Παράδειγμα επικοινωνίας I2C	70
Εικόνα 49 - Παράδειγμα μεταφοράς δεδομένων	71
Εικόνες 50 έως 57 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού	73 έως 77
Εικόνα 58 - Θύρα υποδοχής κάρτας micro sd	77
Εικόνα 59 έως 60 - Εγκατάσταση λογισμικού	78
Εικόνα 61 έως 66 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi	79 έως 82
Εικόνα 67 έως 69 - Διαδικασία σύνδεσης μέσω Wi-fi	83 έως 84
Εικόνα 70 - Σύνδεση Logitech Webcam c270	87
Εικόνα 71 έως 72 - Ενεργοποίηση της κάμερας μέσω Raspberry Pi Configuration	88
Εικόνα 73 - Terminal	89
Εικόνα 74 - Σχεδιάγραμμα σύνδεσης Rapsberry pi με MPU6050	91
Εικόνα 75 - Σύνδεση του ολοκληρωμένου L298N στο Raspberry pi	92
Εικόνα 76 - Δημιουργία ζιγκ-ζαγκ με δεξιά και αριστερή γωνία 90° μοιρών	94
Εικόνα 77 - Πρώτη δοκιμή του οχήματος στην πίστα	94
Εικόνα 78 - Αλλαγή πίστας και πρώτη βόλτα	95
Εικόνα 79 - Προσθήκη σημάτων	95
Εικόνα 80 - Το σήμα “Wait”	96
Εικόνα 81 - Το τελικό σημείο της διαδρομής το σήμα “Stop”	96
Εικόνα 82 - Συνδέσεις jumpers στις GPIO εξόδους του Raspberry Pi	97
Εικόνα 83 - Συνδέσεις μοτέρ	97
Εικόνα 84 - Τελική μορφή του πρώτου σασί	98
Εικόνα 85 - Τα κομμάτια του σασί	98
Εικόνα 86 - Το κάτω επίπεδο με το Mpu6050 και το L298n	99
Εικόνα 87 - Οι έξοδοι Gpio του Raspberry Pi	99
Εικόνα 88 - Ολοκληρωμένα τα δύο επίπεδα	100
Εικόνα 89 - Πριν την τοποθέτηση της κάμερας	100
Εικόνα 90 - Η τελική μορφή του δεύτερου σασί	101



# ΚΕΦΑΛΑΙΟ 1

## 1.1 Ρομπότ και Ρομποτική

### 1.1.1 Εισαγωγή

Η **Ρομποτική** είναι ο κλάδος της επιστήμης που μελετά τις μηχανές εκείνες που δημιουργήθηκαν ώστε να αντικαταστήσουν τον άνθρωπο στην εκτέλεση μιας εργασίας. Η ρομποτική αποτελεί ένα μείγμα επιστημών από Ηλεκτρονική ,Μηχανολογία , Πληροφορική ,τον Αυτοματισμό (Σ.Α.Ε) και τις Τηλεπικοινωνίες . Αυτό επιτυγχάνεται σε συνδυασμό μηχανισμών πχ βραχίονες και αισθητήρων (κίνησης, ήχου, φωτός κ.α ) .Το μέλλον της ρομποτικής είναι “μεγάλο” και μένει να δούμε αν με την εξέλιξη της σε συνδυασμό με την τεχνητή νοημοσύνη οδηγήσει σε εικόνες που έχουμε δει μόνο σε ταινίες.



Εικόνα 1 - Ρομπότ στις Ταινίες

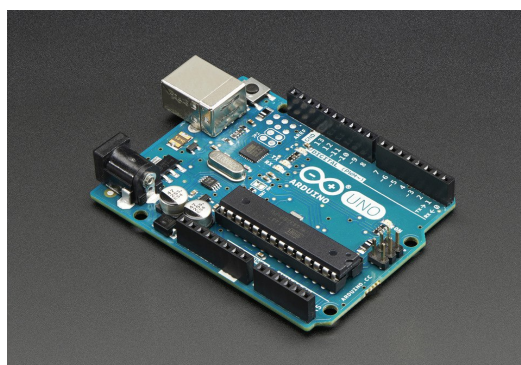
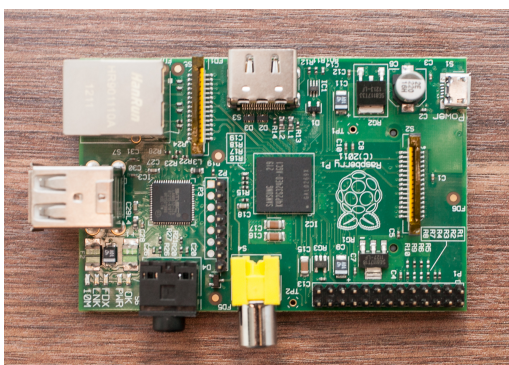


### 1.1.2 Τι Είναι Ρομπότ ;

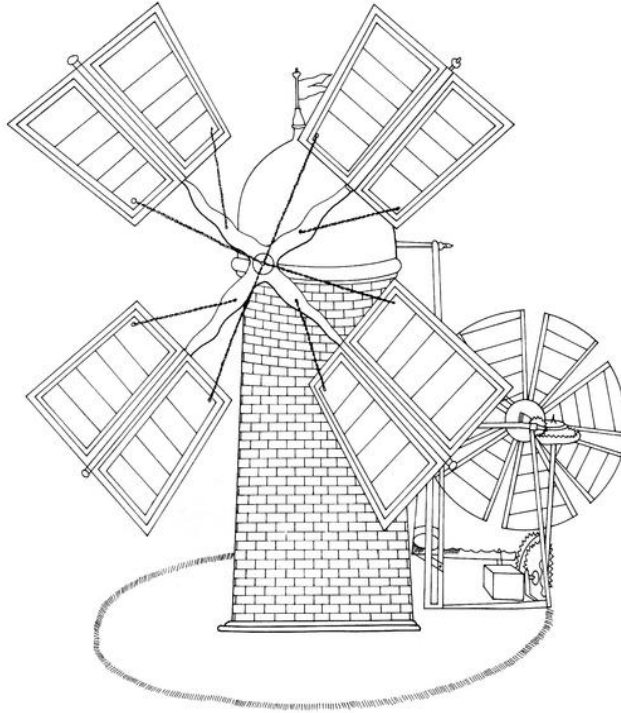
Αν η ρομποτική είναι η επιστήμη , **Ρομπότ** είναι η συσκευή - μηχανή που έχει κατασκευαστεί και προγραμματιστεί για να αντικαταστήσει τον άνθρωπο σε κάποιες από τις εργασίες του . Πιο συγκεκριμένα ένας τέτοιος μηχανισμός μας δίνει την δυνατότητα να μεταφέρουμε μεγάλα φορτία είτε να πραγματοποιήσουμε εργασίες που απαιτούν λεπτούς χειρισμούς .Ο άνθρωπος αδυνατούσε να κάνει τέτοιου είδους εργασίες. Η δυνατότητα αυτή οδήγησε σε αύξηση της παραγωγικότητας βελτίωση της ποιότητας και σε αύξηση του κέρδους των επιχειρήσεων .Ο όρος ρομπότ παρουσιάστηκε πρώτη φορά σε θεατρική παράσταση επιστημονικής φαντασίας και όχι στα χαρτιά κάποιου εφευρέτη .Πιο συγκεκριμένα το έργο ήταν του Τσέχου συγγραφέα **Κάρελ Τσάπεκ** το **1921** και προέρχεται από τη σλαβική λέξη *robo* που σημαίνει εργασία .Τα κύρια μέρη ενός ρομπότ παρουσιάζονται παρακάτω.

### Σύστημα Ελέγχου (CONTROL SYSTEM)

Ο άνθρωπος όπως και πολλά άλλα ζώα αντιλαμβάνονται τι συμβαίνει και πράττουν αναλόγως θα λέγαμε ότι συλλέγουν δεδομένα αυτή είναι η βασική αρχή ανατροφοδότησης .Πρώτη φορά εμφανίστηκε σε μηχανή ο μηχανισμός αυτός το **1745** από τον **Έντμαντ Λι** (Edmund Lee) ο οποίος αυτοματοποίησε ένα ανεμόμυλο ,τοποθετώντας δυο μικρότερους οι οποίοι ανάλογα με την κατεύθυνση του αέρα μετακινούσαν έναν άξονα του “κύριου” ανεμόμυλου .Το σύστημα ελέγχου ή μονάδα επεξεργασίας του ρομπότ χρησιμοποιεί ανατροφοδότηση ακριβώς όπως το κάνει ο ανθρώπινος εγκέφαλος .Στα ρομπότ όμως δεν έχουμε κάποια συλλογή νευρώνων, ο εγκέφαλος ενός ρομπότ αποτελείται από ένα τσιπ πυριτίου που ονομάζεται κεντρική μονάδα επεξεργασίας η γνωστή σε όλους μας CPU. Οι εγκέφαλοί μας αποφασίζουν τι πρέπει να κάνουν και πώς να αντιδρούν στον κόσμο με βάση την ανατροφοδότηση από τις πέντε αισθήσεις μας. Η CPU ενός ρομπότ κάνει το ίδιο πράγμα με βάση δεδομένα που συλλέγονται από συσκευές που ονομάζονται αισθητήρες.



Εικόνα 2 - Raspberry Pi, Εικόνα 3 - Arduino



**Εικόνα 4 - Ανεμόμυλος του Edmund Lee**

### **Αισθητήρες (Sensors)**

Τα ρομπότ λαμβάνουν ανατροφοδότηση από αισθητήρες που μιμούνται ανθρώπινες αισθήσεις. Αυτά μπορεί να είναι κάμερες, φωτο-εξαρτώμενες αντιστάσεις ώστε να γίνουν αντιληπτές οι κινήσεις, αισθητήρες θερμοκρασίας, αισθητήρες ήχου, απόστασης, επιταχυνσιόμετρα, γυροσκόπια, πυξίδες. Ορισμένα ρομπότ έχουν ακόμη άγγιγμα, γεύση και οσμή. Η CPU του ρομπότ ερμηνεύει σήματα από αυτούς τους αισθητήρες και προσαρμόζει ανάλογα τις ενέργειές του.



Εικόνα 5 - Αισθητήρες

### Ενεργοποιητές (Actuators)

Πρακτικά μιλάμε για τους κινητήρες που βρίσκονται στο ρομπότ. Το σασί συνήθως είναι κατασκευασμένο από πλαστικό ή κάποιο μέταλλο είτε κράμα μετάλλων μέσα στο οποίο υπάρχουν οι ενεργοποιητές οι οποίοι πραγματοποιούν μίμηση των ανθρώπινων μυών. Τα πιο απλά αποτελούνται από έναν βραχίονα με συγκεκριμένες κινήσεις (Βαθμούς ελευθερίας) και έχουν σαν κατάληξη κάποιο εργαλείο π.χ. κολλητήρι. Τα ανθρωποειδή ρομπότ έχουν πολλούς ενεργοποιητές ώστε να καταφέρουν να μιμηθούν κάθε ανθρώπινη κίνηση.

### Τροφοδοσία (Power Supply)

Η τροφοδοσία ουσιαστικά θα δώσει ζωή στο ρομποτικό μας σύστημα. Τα φυτά παίρνουν την απαιτούμενη ενέργεια από το χώμα και το νερό κάτι αντίστοιχο γίνεται και με τους ανθρώπους που ο οργανισμός μετατρέπει την τροφή σε ενέργεια. Στα ρομποτικά συστήματα αυτή την “δουλειά” αναλαμβάνει ο ηλεκτρισμός. Αν μιλάμε για κάποιον σταθερό βραχίονα σε ένα εργοστάσιο κατευθύνει στο ρεύμα, αν είναι κινούμενο χρησιμοποιείται μπαταρία ενώ στην περίπτωση των δορυφόρων έχουμε ηλιακά πάνελ.

### Εργαλείο Τελικής Δράσης (END EFFECTORS)

Στην άκρη του βραχίονα τοποθετούμε το εργαλείο που είναι κατάλληλο για την εργασία που θα πραγματοποιήσουμε. Αυτό μπορεί να είναι ένα κατσαβίδι, ένα εργαλείο λείανσης κάποιο πιστόλι βαψίματος ή κάποιο κολλητήρι. Αν το ρομποτικό μας σύστημα προορίζεται για να πιάσει και να μεταφέρει κάποιο αντικείμενο τότε πιθανό να έχει κάποια αρπάγη ή κάποιο σύστημα που λειτουργεί με κενό αέρος.

### 1.1.3 Κατηγορίες των ρομπότ

- Βιομηχανικά Ρομπότ
- Ιατρικά
- Οικιακά
- Στρατιωτικά
- Κινούμενα

Τα **Βιομηχανικά ρομπότ** ή ρομπότ σταθερής βάσης χρησιμοποιούνται σε κάποιο εργοστάσιο. Τις περισσότερες των περιπτώσεων είναι ρομποτικοί βραχίονες με συγκεκριμένους συνδέσμους και αρθρώσεις (Βαθμοί Ελευθερίας). Παραδείγματα των εργασιών που μπορούν να πραγματοποιήσουν είναι η συγκόλληση, το βάψιμο, να μεταφέρουν βαριά αντικείμενα κ.α.

Τα **Ιατρικά Ρομπότ** χρησιμοποιούνται κατά κύριο λόγο σε χειρουργεία με αποτέλεσμα λεπτές επεμβάσεις να πραγματοποιούνται με μεγάλη ακρίβεια είτε απομακρυσμένα (Τηλεϊατρική). Η χρήση τους όμως δεν περιορίζεται στα χειρουργεία και περιλαμβάνει ρομπότ αποκατάστασης, φαρμακευτικά κ.α.

**Οικιακά Ρομπότ** είναι τα ρομπότ που θα συναντήσουμε σ' ένα νοικοκυριό. Αυτά αποτελούν αυτοματοποιημένες ηλεκτρικές σκούπες, σε καθαρισμό πισίνας, γυάλισμα δαπέδου και άλλες μικροδουλειές που αυτοματοποιούνται .

**Στρατιωτικά Ρομπότ** είναι μη επανδρωμένα ρομπότ τα οποία χρησιμοποιούνται ως ναρκαλιευτές, drones που στέλνονται για αναγνώριση και για ερευνητικούς σκοπούς.



Εικόνα 6 - Ρομπότ Ναρκαλιευτής, Εικόνα 7 - Drone

**Κινούμενα ρομπότ** ονομάζουμε τα ρομποτικά συστήματα τα οποία έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η κίνηση πραγματοποιείται σε απλές περιπτώσεις με τροχούς και άλλες πιο σύνθετες π.χ. προπέλες, μηχανικά πόδια κ.α . Οι υποκατηγορίες είναι οι εξής :

- **AGVs:** τα AGVs (Automatic Guided Vehicles) είναι τα ρομπότ που έχουν συγκεκριμένη τροχιά που αυτή καθορίζεται είτε με καλώδια στο έδαφος είτε μέσω πομπών.
- **Αυτόνομα Έντροχα Ρομπότ:** είναι ρομπότ με υψηλό βαθμό αυτονομίας δηλαδή δεν κρίνεται απαραίτητη η συνεχής επίβλεψη και εκτελούν εργασίας αυτόνομα.
- **Βαδίζοντα Ρομπότ:** είναι ρομπότ τα οποία χρησιμοποιούν μηχανικά πόδια για την κίνηση τους .Αυτό κάνει ευκολότερο την αποφυγή εμποδίων και την βάδιση σε ανώμαλα εδάφη .Τις περισσότερες φορές είναι ανθρωπόμορφα ή σε εφαρμογές με περισσότερα πόδια μπορεί έχουν την κίνηση και τον σχηματισμό μιας αράχνης.
- **ROVs:** τα ROVs (Remotely Operated Vehicles) είναι μη επανδρωμένα υποβρύχια ρομπότ. Δεν έχουν μεγάλο βαθμό αυτονομίας και είναι συνδεδεμένα μέσω καλωδίων με κάποιο μητρικό πλοίο .Έχουν σχήμα κουτιού και κινούνται σε χαμηλές ταχύτητες .
- **Εναέρια ρομπότ (UAV):** πρόκειται για μη επανδρωμένα ιπτάμενα ρομπότ, όπως ελικόπτερα και αεροπλάνα. Η χρήση τους γίνεται όλο και μεγαλύτερη με το πέρασμα των χρόνων. Αρχικά τα βλέπαμε κυρίως σε στρατιωτικούς σκοπούς αλλά τα τελευταία χρόνια τα τετρακόπτερα (quadcopters) έχουν γίνει αρκετά δημοφιλή καθώς αποτελούν σχετικά φθηνές λύσεις για χόμπι, κινηματογράφηση, μεταφορά προϊόντων κ.α.

#### 1.1.4 Παρελθόν,παρόν και μέλλον ρομποτικής

Το παρελθόν της ρομποτικής ορίζεται αρκετά χρόνια π.Χ η πρώτη αναφορά γίνεται από τον Έλληνα φιλόσοφο Αριστοτέλη το 320 π.Χ είχε πει την φράση «Αν, κάθε εργαλείο μπορούσε να εκπληρώσει το έργο του με μια εντολή...αν οι αργαλειοί ύφαιναν από μόνοι τους και το πλήκτρο χτύπαγε από μόνο του την κιθάρα, τότε ούτε οι μάστορες θα χρειάζονταν βοηθούς, ούτε οι αφέντες δούλους» πόσο κοντινό μοιάζει στην εποχή που ζούμε. Σ' αυτήν την παράγραφο θα αναλύσουμε την ιστορία ,το παρόν και το μέλλον της ρομποτικής.

**278 – 212 π.Χ:** Ο Αρχιμήδης μπορεί να μην εφηύρε την ρομποτική αλλά εφάρμοσε αρκετούς μηχανισμούς που χρησιμοποιούνται ακόμα και σήμερα στον κλάδο.

**~270 π.Χ:** Ο Κτησίβιος ο Αλεξανδρεύς ήταν ένας σπουδαίος μηχανικός και εφευρέτης ,η εφεύρεση του **Υδραυλικό Ωρολόγιο** αποτελεί ένα θαύμα του αυτοματισμού για την εποχή. Πρακτικά πρόκειται για μια δεξαμενή με μια ακριβή τρύπα στον πυθμένα της όπου θα χρειαζόταν 24 ώρες για να αδειάσει το περιεχόμενο της μ' αυτόν τον τρόπο εξασφάλιζε με ακρίβεια την μέτρηση της ώρας.

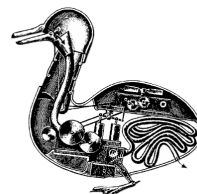
**~77-100 π.Χ:** Το 1901, ανάμεσα στα νησιά της Κρήτης και των Κυθήρων, βρέθηκε

ένας μηχανισμός που θα μπορούσε να θεωρηθεί ο πρώτος μηχανικός υπολογιστής. Η συσκευή αποτελείται από ένα πολύπλοκο μείγμα ταχυτήτων που υπολογίζει τη θέση του ήλιου, το φεγγάρι ή άλλα ουράνια σώματα. Η συσκευή χρονολογείται περίπου 2000 χρόνια πριν, θεωρείται ελληνικής καταγωγής και του δόθηκε το όνομα "**Μηχανισμός Των Αντικυθήρων**".

**1495:** Ο **Λεονάρντο ντα Βίντσι** σχεδίασε το πρώτο ανθρωποειδές ρομπότ. Οι λειτουργίες που είχε σχεδιάσει ο εφευρέτης ήταν να μπορεί να καθίσει, να κουνήσει τα χέρια, το κεφάλι και ανοιγοκλείσει το σαγόνι του.

**1645:** Ο Μπλεζ Πασκάλ εφηύρε μια υπολογιστική μηχανή για να βοηθήσει τον πατέρα του με τους φόρους. Η συσκευή ονομάστηκε Pascaline και κατασκευάστηκαν περίπου 50 Pascalines. Λίγες μπορούν να βρεθούν σε μουσεία όπως στο μουσείο "**Τεχνών και Επαγγελματιών**" στο Παρίσι.

**1709:** Το πιο διάσημο έργο του Jacques de Vaucanson ήταν αναμφισβήτητα "Ηπάτια". Αυτή η μηχανική συσκευή θα μπορούσε να πλέξει τα φτερά της, να φάει και να αφομοιώσει τους κόκκους. Κάθε πτέρυγα αποτελούνταν με πάνω από τετρακόσια κινούμενα μέρη και ακόμα και σήμερα η κατασκευή της παραμένει κάτι μυστήριο. Η αρχική πάπια έχει εξαφανιστεί.



Εικόνα 8 - Πάπια

**1801:** Ο μηχανικός – υφαντουργός Ζοζέφ Ζακάρ εφηύρε μια μηχανή (ουσιαστικά αργαλειό) που θα μπορούσε να προγραμματιστεί ώστε να δημιουργεί πολύπλοκα σχέδια που θα μπορούσαν να εκτυπωθούν σε ύφασμα ή ιστό. Ο αργαλειός του Ζακάρ έφερε επανάσταση στη κλωστοϋφαντουργία και οι βασικές αρχές του εξακολουθούν να χρησιμοποιούνται και σήμερα.

**1937-1938:** Η εταιρία Westinghouse δημιούργησε τον Ελέκτρο ένας ανθρωποειδές ρομπότ το οποίο μπορεί να περπατήσει να μιλήσει και να καπνίσει .

**1941:** Ο επιστημονικής φαντασίας συγγραφέας Ισαάκ Ασίμοφ χρησιμοποίησε το όρο "robotics" για να περιγράψει την τεχνολογία των ρομπότ .

**1942:** Ο πρώτος "προγραμματιζόμενος" μηχανισμός, ψεκαστήρας βαφής, σχεδιάστηκε από τον Willard Pollard και Harold Roselund για την εταιρεία DeVilbiss.

**1948:** Ο W. Grey Walter κατασκεύασε το πρώτο του ρομπότ Elmer and Elsie, γνωστό και ως ρομπότ χελώνα. Τα ρομπότ αυτά έβρισκαν τον σταθμό επαναφόρτισης μόλις η μπαταρίας τους ήταν σε χαμηλή στάθμη.



Εικόνα 9 - Ρομπότ Χελώνα

**1950:** Ο Alan Turing προτείνει μια δοκιμή για να διαπιστώσει αν μια μηχανή έχει πραγματικά τη δύναμη να σκεφτεί. Για να περάσει η δοκιμή, μια μηχανή πρέπει να είναι αδιάριετη από τον άνθρωπο κατά τη διάρκεια της συνομιλίας. Έχει γίνει γνωστή ως η «δοκιμή Turing».

**1954:** Ο George Devol σχεδίασε το πρώτο πραγματικά προγραμματιζόμενο ρομπότ που το ονόμασε UNIMATE (Universal Automation). Το 1956 μαζί με τον

Joseph Engelberger ίδρυσαν την πρώτη εταιρία παραγωγής ρομπότ (UNIMATION) . Ο Joseph Engelberger ονομάστηκε πατέρας της ρομποτικής .

**1957:** Ιστορία γράφτηκε στις 4 Οκτωβρίου του 1957 όταν η Σοβιετική Ένωση εκτόξευσε επιτυχημένα τον πρώτο αυτόνομο τεχνητό δορυφόρο ονόματι **Sputnik I**.

**1964:** Ο υπολογιστής I.B.M 360 έγινε ο πρώτος που βγήκε σε μαζική παραγωγή.

**1968:** Η SRI κατέυασε ένα κινούμενο ρομπότ με όνομα “Shakey”;εξοπλισμένο με σύστημα “οράσεως” το οποίο ελέγχεται από έναν υπολογιστή σε μέγεθος δωματίου .

**1969:** Ο Victor Scheinman δημιούργησε τον πρώτο ηλεκτρικό ελεγχόμενο από υπολογιστή ρομποτικό βραχιόνα το ονόμασε Stanford Arm.

**1973:** Ο Ichiro Kato δημιούργησε τον WABOT I, ο οποίος ήταν το πρώτο πλήρους κλίμακας ανθρωπόμορφο Ρομπότ στον κόσμο. Είχε ένα σύστημα ελέγχου των άκρων, της όρασης και της συνομιλίας.

**1979:** Το Stanford Cart διέσχισε ένα δωμάτιο με εμπόδια χωρίς ανθρώπινη επέμβαση. Είχε μία κάμερα τηλεόρασης τοποθετημένη σε μια ράγα που έβγαζε φωτογραφίες από πολλαπλές γωνίες και η μετάδοση γινόταν σε έναν υπολογιστή. Ο υπολογιστής ανέλυσε την απόσταση μεταξύ του καροτσιού και των εμποδίων.



Εικόνα 10 - Stanford Cart

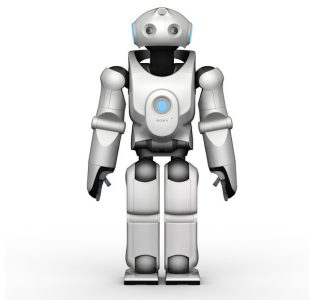
**1985:** Δημιουργήθηκε από τη General Robotics Corp. το RB5X ήταν ένα προγραμματιζόμενο ρομπότ που ήταν εξοπλισμένο με υπέρυθρους αισθητήρες, απομακρυσμένη μετάδοση ήχου / εικόνας, αισθητήρες επαφής και μια συσκευή σύνθεσης φωνής. Το λογισμικό του επέτρεπαι να μάθει για το περιβάλλον του

**1989:** Το WL12RIII που αναπτύχθηκε από την Kato Corporation ήταν το πρώτο ρομπότ που ήταν σε θέση να περπατάει ενώ παράλληλα ισορροπούσε με την κίνηση του κορμού. Τέλος μπορούσε να ανεβοκατεβαίνει σκάλες καθώς και να εκτελεί ένα βήμα κάθε 0.64 δευτερόλεπτα.

**1993:** Ο Dante διερεύνησε το Mt. Erebus στην Ανταρκτική. Το ρομπότ με τα πόδια με 8 πόδια αναπτύχθηκε στο πανεπιστήμιο Carnegie-Mellon. Ωστόσο, η αποστολή απέτυχε. Τελικά ο Dante II διερεύνησε στη συνέχεια το Mt. Spurr στην Αλάσκα το 2004 ήταν μια βελτιωμένη εκδοχή του Dante I.

**1997:** Ο “Deep Blue” υπερυπολογιστής της IBM κέρδισε τον πρωταθλητή Garry Kasparov σε έναν αγώνα σκάκι. Αυτή ήταν η πρώτη φορά που μια μηχανή νικάει ένα μεγάλο πρωταθλητή σκάκι.

**2000:** Η Sony παρουσίασε τα ρομπότ Sony Dream (SDR) στο Robodex. SDR μιλάμε για ρομπότ που ήταν σε θέση να αναγνωρίζουν 10 διαφορετικά πρόσωπα, εκφράσουν το συναίσθημα μέσω του λόγου και της γλώσσας τους σώματος, και να περπατήσουν σε επίπεδες καθώς και ακανόνιστες επιφάνειες.



Εικόνα 11 - Ρομπότ Sony Dream (SDR)

**2002:** Η Honda δημιούργησε το Advanced Step in Innovative Mobility (ASIMO). Ένα ρομπότ προσωπικός βοηθός. Έχει την δυνατότητα να αναγνωρίζει το πρόσωπο, τη φωνή του ιδιοκτήτη και το όνομα. Μπορεί να διαβάζει μηνύματα ηλεκτρονικού ταχυδρομείου και να μεταδώσει βίντεο από τη φωτογραφική μηχανή σε έναν ηλεκτρονικό υπολογιστή .

**2005:** Το Κορεατικό Ινστιτούτο Επιστήμης και Τεχνολογίας (KIST) δημιούργησε το “HUBO” και ισχυρίζεται ότι είναι το πιο έξυπνο κινητό ρομπότ στον κόσμο. Αυτό το ρομπότ είναι συνδεδεμένο με έναν υπολογιστή μέσω ασύρματης σύνδεσης υψηλής ταχύτητας.

**2007:** Το i-LIMB πρόκειται για το πρώτο εμπορικό προσθετικό χέρι του κόσμου, που δημιουργήθηκε από την Touch Bionics, ήταν τόσο πολλαπλών άρθρων (κάθε δάκτυλο είναι ικανό για τη δική του κίνηση) όσο και ικανό για λαβή ισχύος (για τη συγκράτηση μεγαλύτερων αντικειμένων). Σε ένα μόνο χρόνο, περισσότεροι από 200 ασθενείς είχαν εφοδιαστεί με χέρια i-LIMB.

**2013:** Το Kirobo είναι ένα ρομπότ αστροναύτης που μιλάει. Σε μόλις 13 ίντσες ύψος, ο Kirobo έγινε ο πρώτος συνομιλητής ανθρωποειδούς αστροναύτη στον κόσμο, είπε τις πρώτες λέξεις στα ιαπωνικά στις 21 Αυγούστου ενώ βρισκόταν στον Διεθνή Διαστημικό Σταθμό (ISS). Το Kirobo, το οποίο σχεδιάστηκε από το Πανεπιστήμιο του Τόκιο, την Toyota και την διαφημιστική εταιρεία Dentsu, για να περιμένει την άφιξη του αστροναύτη Koichi Wakata. Ο Kirobo είναι ένα ρομπότ με συναισθήματα.

Όπως είδαμε η ρομποτική σαν επιστήμη μπορεί να είναι πρόσφατη αλλά αυτοματισμούς και εφευρέσεις που αποτελούν την βάση της χρονολογούνται πριν



2000 χρόνια. Η ανάπτυξη ηλεκτρονικών και της μηχανικής μας έχουν δώσει εφευρέσεις που έχουν αλλάξει την ζωή μας. Έχουμε δει ρομπότ με τεχνητή νοημοσύνη που έχουν την δυνατότητα μάθησης, ρομπότ που βοηθούν τις δουλειές του σπιτιού, βιονικά άκρα, ρομπότ ναυαγοσώστες, drones που μεταφέρουν δέματα κ.α

Τι μας επιφυλάσσει το μέλλον δεν μπορούμε να το πούμε με ακρίβεια. Η εξέλιξη της τεχνολογίας πάντως μας προϊδεάζει πως τα ρομπότ θα αναπτύξουν υψηλή τεχνητή νοημοσύνη και θα μπορούσαν μελλοντικά να υποκαταστήσουν τον άνθρωπο και σε κρίσιμες θέσεις εργασίας. Το ερώτημα είναι πόσο σύντομα θα μπορούσε να γίνει κάτι τέτοιο. Σύμφωνα με την Μέριλ Λιντς, αναλύτρια από την Τράπεζα της Αμερικής, περί το 2025 περίπου το 45% των εργασιών στον βιομηχανικό τομέα αναμένεται να εκτελείται από ρομπότ. Σήμερα το ίδιο ποσοστό ανέρχεται στο 10%.

Είναι αποκαλυπτικό ότι υπάρχουν περισσότερα από δέκα εκατομμύρια ρομπότ-σκούπερ που σκουπίζουν πατώματα σε όλο τον κόσμο. Επίσης, είναι γνωστό ότι τα σημερινά αυτοκίνητα κατασκευάζονται με τη βοήθεια τεραστίων ρομπότ. Ρομπότ κατασκευάζουν και τα ολοκληρωμένα κυκλώματα, που είναι η βάση των υπολογιστών και των κινητών τηλεφώνων. Υπάρχουν, ακόμα, και άλλες, πολύ εξειδικευμένες δραστηριότητες που εκτελούνται από ρομπότ, όπως η εξερεύνηση του πλανήτη Άρη και η εξουδετέρωση εκρηκτικών μηχανισμών. Χωρίς τα ρομπότ, ο ρυθμός παραγωγής αγαθών θα ήταν αργός, πολλές ανακαλύψεις θα ήταν αδύνατες, ενώ πολλά αγαθά καθημερινής χρήσης, όπως τα αυτοκίνητα, οι προσωπικοί υπολογιστές και τα κινητά τηλέφωνα, θα κόστιζαν μια περιουσία.

Χάρη στην τεχνολογία πολλές πτυχές του εργασιακού βίου έχουν βελτιωθεί σημαντικά, εφόσον έχει μειωθεί κατά πολύ ο χρόνος δραστηριοτήτων που άλλοτε ήταν εξαιρετικά χρονοβόρες, έχει ελαττωθεί ο χειρωνακτικός μόχθος κι έχει επιτευχθεί ένας ιδιαίτερα αποδοτικός καταμερισμός των εργασιών. Αποτέλεσμα αυτών είναι, εύλογα, η αύξηση της παραγωγικότητας και η ενίσχυση της οικονομικής ανάπτυξης.

Επίσης οι αλλαγές στον τομέα της υγείας δεν μπορούν να μην ληφθούν υπόψιν. Η τεχνολογική εξέλιξη έχει βρει άριστη εφαρμογή στο χώρο της υγείας επιτρέποντας τη βελτίωση των χειρουργικών επεμβάσεων κατά τέτοιο τρόπο ώστε να γίνονται πιο αποτελεσματικές, αλλά λιγότερο παρεμβατικές, γεγονός που μειώνει τον αναγκαίο χρόνο ανάρρωσης.

Υπάρχουν μειονεκτήματα που μπορεί να παρουσιάσει η ανάπτυξη της ρομποτικής με πρώτο και κύριο την μείωση των θέσεων εργασίας. Η εξέλιξη της

τεχνολογίας έχει επιτρέψει αφενός την αυτοματοποίηση στη βιομηχανική και βιοτεχνική παραγωγή κι αφετέρου τη μείωση των γραφειοκρατικών αναγκών, με άμεσο αποτέλεσμα τον περιορισμό των αναγκών σε ανθρώπινο δυναμικό. Οι μηχανές παραγωγής και οι ηλεκτρονικοί υπολογιστές έχουν καταστήσει περιττές χιλιάδες θέσεις εργασίας, εφόσον πλέον διαδικασίες που κάποτε απαιτούσαν την ανθρώπινη μεσολάβηση τελούνται πολύ πιο αποτελεσματικά με αυτοματοποιημένο τρόπο. Αυτό βέβαια θα μπορούσε να αντικρούσει κάποιος λέγοντας ότι δημιουργούνται και νέα επαγγέλματα.

Αίσθημα αλλοτρίωσης από την εργασία. Η αυτοματοποίηση των εργασιών εκείνων που σχετίζονται με την παραγωγή αγαθών προκαλεί ένα έντονο και ψυχοφθόρο συναίσθημα αλλοτρίωσης, εφόσον ο εργαζόμενος καλείται να πραγματοποιεί για ώρες καθημερινά τις ίδιες τυποποιημένες ενέργειες. Χάνεται, έτσι, το αίσθημα της προσωπικής σύνδεσης με το τελικό προϊόν και το μόνο που απομένει είναι η ανία της διαρκούς και μηχανικής επανάληψης.

Η ρομποτική είναι συνδεδεμένη άρρηκτα με την τεχνολογία. Όποτε πρέπει να προσέξουμε την χρήση της. Αν αναλογιστούμε πως μεγάλες εταιρείες κατασκευής λογισμικού έχουν ανακοινώσει ότι βρίσκεται σε εξέλιξη η κατασκευή προγραμμάτων τα οποία θα αντικαταστήσουν οδηγούς, γκαρσόνια, ακόμη και νοσοκόμους, συμπεραίνουμε πως σε είκοσι περίπου χρόνια από σήμερα τα ρομπότ θα αποτελούν αναπόσπαστο κομμάτι της ζωής μας.



## ΚΕΦΑΛΑΙΟ 2

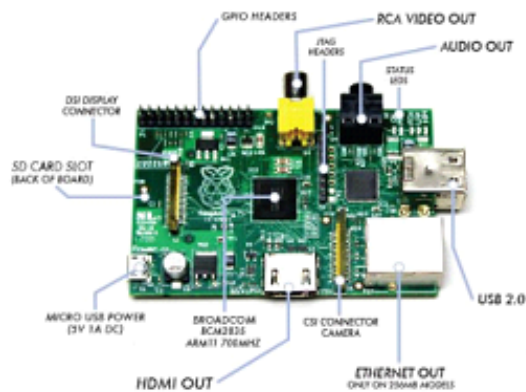
### 2.1 Υλικά Κατασκευής

Σε αυτήν την ενότητα θα περιγραφούν τα υλικά τα οποία χρησιμοποιήθηκαν για την υλοποίηση της πτυχιακής.

- Raspberry Pi πλακέτα (Υπολογιστικό Σύστημα)
- USB Camera
- Micro SD card
- Σάσι (Chassis) με ηλεκτρικούς κινητήρες
- MPU6050 (Γυροσκόπιο , Επιταχυνσιόμετρο)
- L298N ολοκληρωμένο (chip)
- Breadboard
- Θηλυκά και αρσενικά καλώδια (jumpers)
- Εξωτερικό φορτιστή μπαταριών (powerbank) και 9 Volt Μπαταρία
- Αντάπτορας Wi-fi

#### 2.1.1 Raspberry Pi

Το ρόλο του μικροελεγκτή (microcontroller) στην κατασκευή μας παίζει το raspberry pi .



Εικόνα 12 - Τα μέρη του Raspberry Pi

Το Raspberry pi αποτελεί έναν υπολογιστή σε μέγεθος πιστωτικής κάρτας. Αναπτύχθηκε από την Raspberry Pi Foundation και σκοπό είχε την προώθηση της διδασκαλίας της επιστήμης των υπολογιστών σε σχολεία. Είναι μια συσκευή η οποία παρουσιάζει πολλές δυνατότητες και είναι ικανή για λειτουργίες που συνήθως της κάνει ένας προσωπικός υπολογιστής, όπως προβολή ταινιών, παιχνίδια, internet browsing κλπ. Ο λόγος επιλογής αυτού του board ήταν ότι αποτελεί ένα σύγχρονο υπολογιστικό σύστημα με πολλές λειτουργικότητες και χαμηλό κόστος.

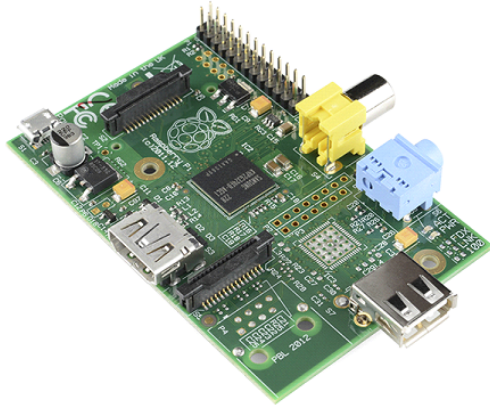
Για την διπλωματική εργασία επιλέξαμε το Raspberry Pi 1 Model B. Το raspberry pi διαθέτει μία SDCard η οποία χρησιμοποιείται για την αποθήκευση αρχείων. Έπισης SDCard είναι απαραίτητη στην boot διαδικασία καθώς όλα τα απαραίτητα αρχεία για την διαδικασία αυτή αποθηκεύονται σε ένα FAT32 partition της κάρτας. Όπως μπορεί να παρατηρήσει κανείς στο raspberry pi, υπάρχει ένας header ο οποίος διαθέτει 26 ακίδες. Πρόκειται για τον P1 header. Συνολικά στο board υπάρχουν ακόμη πέντε headers αλλά δεν είναι ενεργοποιημένοι. Ο P1 header είναι γνωστός και ως expansion header ή gpio connector P1. Σε αυτό τον header πέραν από κάποιες ακίδες τροφοδοσίας 5V, 3.3V και GND, οι υπόλοιπες ακίδες έχουν κάποιες συγκεκριμένες λειτουργίες που επιτρέπουν την επικοινωνία και σύνδεση του raspberry pi με κάποιες άλλες συσκευές. Η τροφοδοσία στο raspberry επιτυγχάνεται μέσω μιας mini usb θύρας που υπάρχει στο board ή μέσω των 5V και GND pins.

## Ιστορία του Rasperry Pi

Οι πρώτοι που συνέλαβαν την ιδέα της δημιουργίας αυτής της πλακέτας της εικόνας ήταν οι Eben Upton και συνεργάτες από το Πανεπιστήμιο του Εργαστηρίου Υπολογιστών του Cambridge, συμπεριλαμβανομένων των Rob Mullins, Jack Lang και ο Alan Mycroft. Η αιτία που οδηγήθηκαν στην δημιουργία της πλακέτας και κατ' επέκταση και του Pi Foundation Raspberry ήταν η μείωση του αριθμού των φοιτητών που εισάγονταν στη σχολή. Οι συγκεκριμένοι φοιτητές και ιδρυτές του Pi Foundation Raspberry σύγκριναν τη κατάσταση που επικρατούσε τότε, το 2006, με το πώς αντιμετώπιζαν τους ηλεκτρονικούς υπολογιστές τη δεκαετία του '90 και αντιλήφθηκαν ότι η αντίληψη των νέων άλλαξε και συνεχώς αλλάζει. Μόλις οι νέες καταστάσεις και απαιτήσεις αξιολογήθηκαν, αντιλήφθηκαν ποιο προϊόν έλειπε από την αγορά και έβαλαν σκοπό για τη δημιουργία του όπου θα ήταν σε προσιτή τιμή ώστε να μπορεί να έχει πρόσβαση σε αυτό ο ενδιαφερόμενος. Σήμερα το Pi Raspberry κοστίζει 5-35 ευρώ μόλις και το πρώτο που κατασκευάστηκε είχε το μέγεθος μιας πιστωτικής κάρτας. Η λειτουργία του μπορούσε να συνδυαστεί μέσω της σύνδεσης του με μια τηλεόραση ή με μια οθόνη αφής.

## Είδη πλακέτας Raspberry Pi

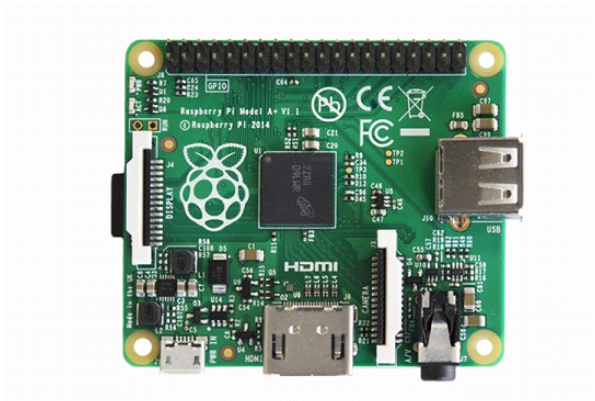
### Type A



Εικόνα 13 - Raspberry Pi Type A

Το μοντέλο είναι παραλλαγή του Raspberry Pi, με μνήμη 256 MB με μία θύρα USB. Το μοντέλο είναι ελαφρύτερο και καταναλώνει λιγότερη ενέργεια από τα υπόλοιπα μοντέλα. Το Λειτουργικό με Linux .Αυτό το μοντέλο σε συνδυασμό με wifi είναι ιδανικό για τους χρήστες που θέλουν απλά ένα media center.

### Type A+



Εικόνα 14 - Raspberry Pi Type A+

Αυτό το Raspberry Pi μοντέλο είναι ένα εξαιρετικά χαμηλού κόστους, έχει μικρό εμβαδόν και αποτελεί βάση για χρήση σε υπολογιστή. Αναπτύχθηκε από το Raspberry Pi Foundation στο Ηνωμένο βασίλειο. Το Raspberry Pi αποσκοπεί να θέσει ξανά τη διασκέδαση σε υπολογιστικά συστήματα μάθησης με την προώθηση της μελέτης της επιστήμης υπολογιστών και εκμάθηση σε σχολικό επίπεδο. Το

μοντέλο ενός Raspberry Pi A+ είναι η γυμνή εκδοχή του μοντέλου B Raspberry Pi, με το Ethernet, μία θύρα USB και μνήμη 256 MB RAM.

Τύπος επεξεργαστή	Broadcom BCM2835 ARMv6 So
Ταχύτητα επεξεργαστή	Μονός Πυρήνας, 700MHz
Μνήμη	256MB
Αποθήκευση	microSD
Πόρτα Ethernet	Δεν περιέχει
USB πόρτα	1 x USB 2.0
GPIO	40pin
Audio	HD κανάλια μέσω HDMI, Στερεοφωνικός ήχος από 3,5mm jack
Video	HDMI, Composite RCA
Power	5volt microUSB, 200mA
Διαστάσεις	65 x 56 x 12mm

### Type B

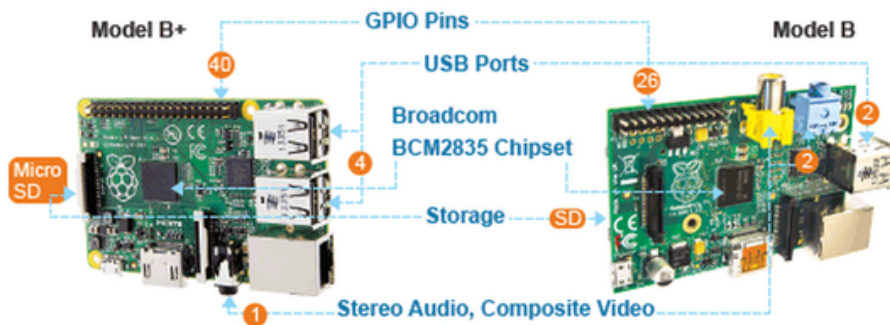


Εικόνα 15 - Raspberry Pi Type B

Τύπος επεξεργαστή	Broadcom BCM2835 ARMv6 SoC
Ταχύτητα επεξεργαστή	Τέσσερις Πυρήνες, 700MHz

Μνήμη	512 mb
Αποθήκευση	microSD
Πόρτα Ethernet	1 x 10/100
USB πόρτα	2 x USB 2.0
GPIO	40pin
Video	HDMI, Composite RCA
Audio	HD κανάλια μέσω HDMI, Στερεοφωνικός ήχος από 3,5mm jack
Power	5volt microUSB, 800mA
Διαστάσεις	85 x 56 x 21mm

### Type B+



Εικόνα 16 - Raspberry Pi Type B+

Το μοντέλο B+ έχει κάποιες διαφορές σε σύγκριση με το μοντέλο B οι οποίες παρουσιάζονται στον παραπάνω εικόνα .

- Περισσότερες GPIO υποδοχές
- Περισσότερες USB θύρες
- Καλύτερη τεχνολογία Micro SD
- Μειωμένη κατανάλωση ρεύματος μεταξύ 0,5 W και 1W
- Καλύτερη ποιότητα ήχου

Αυτό το μοντέλο συνιστάται πιο πολύ για αυτούς που θέλουν να μάθουν ηλεκτρονικά γιατί προσφέρει μεγαλύτερη ευελιξία, είναι πιο χρήσιμο για



ενσωματωμένα προγράμματα και έργα που απαιτούν χαμηλή ισχύ, και ένα τελευταίο είναι κατάλληλο γιατί μας παρέχει περισσότερες θύρες USB με αποτέλεσμα την επεκτασιμότητα.

Παρακάτω θα αναλύσουμε το Raspberry Pi model B το οποίο επιλέξαμε για την πτυχιακή μας εργασία: Η πλατφόρμα είναι εξοπλισμένη με την πλακέτα της Broadcom “BCM2835” διαστάσεων 88,6mm x 53,98mm και συνολικό βάρος 45g, περιέχει τον επεξεργαστή “ARM1176JZ-Score” της οικογένειας ARM11 με συχνότητα 700MHz και κάρτα γραφικών την “Video Core IV” της Broadcom με συχνότητα 250MHz. Το μοντέλο τύπου B έχει μνήμη τύπου SDRAM 512MB. Διαθέτει μια θύρα HDMI και μια RCA για σύνδεση οθόνης και μια 3,5mm jack audio για θύρα εξόδου ήχου. Τέλος, διαθέτουν μια κάρτα υποδοχής (SD CARD) στην κάτω πλευρά της πλακέτας, η οποία χρησιμοποιείται για να φορτωθεί το λειτουργικό σύστημα, και δύο υποδοχές καλωδιωτικής. Τα Led που περιέχονται είναι τα εξής ACT, PWR, FDX, LNK, 100. Το ACT Led αναβοσβήνει όταν υπάρχει δραστηριότητα (διάβαση λειτουργικού συστήματος) μεταξύ SD card και του raspberry pi, το Led PWR μένει ανοιχτό μόλις το raspberry pi τροφοδοτείται με 5volt και κλείνει μόλις η τροφοδοσία υποβιβαστεί κάτω από 4,65volt, το Led PDX ανάβει όταν ιδρυθεί η ενσύρματη σύνδεση μεταξύ του raspberry pi και ενός σημείου πρόσβασης (π.χ. laptop, router), το Led LNK ανάβει όταν συνδέσουμε το καλώδιο του Ethernet από το raspberry pi με ένα σημείο πρόσβασης και το Led 100 ανάβει όταν η μεταφορά δεδομένων γίνεται με 100Mbits ανά δευτερόλεπτο.

Με τα χρόνια η ανάπτυξη της τεχνολογίας αλλά και οι ανάγκες του καταναλωτικού κοινού καθώς και η μεγάλη αποδοχή η οποία είχε η πλατφόρμα έφερε νέα μοντέλα. Τα μοντέλα που παρουσιάζει η εταιρία είναι Raspberry pi 1 (Model A , Model B & Model B+) που αναλύσαμε παραπάνω. Το Raspberry pi 2 κυκλοφόρησε το Φεβρουάριο του 2015 και έχει την διπλάσια μνήμη από τον προκάτοχο του 1GB και ταχύτερο επεξεργαστή ARMv7 στα 900MHz και τέσσερις θύρες usb. Πιο πρόσφατη είναι η κυκλοφορία του Raspberry pi 3 τον Φεβρουάριο του 2016 που είναι σαφώς βελτιωμένο. Το Soc είναι Broadcom BCM2837 και ο επεξεργαστής είναι ο τετραπύρηνος ARM Cortex-A53 στα 1.2GHz η μνήμη παραμένει στα 1GB αλλά εδώ θα δούμε έχει ενσωματωμένο bluetooth και Wifi LAN κάτι που ήταν έξτρα στα προηγούμενα μοντέλα. Επίσης υπάρχει η δυνατότητα εγκατάστασης μίας ελεύθερης έκδοσης των Windows 10. Τέλος έχει κυκλοφορήσει το Raspberry pi zero τον Νοέμβριο του 2015 μια πολύ φτηνή (5 \$) έκδοση. Το μικρό του μέγεθος το κάνει να ξεχωρίζει. Τα τεχνικά του χαρακτηριστικά είναι ο επεξεργαστής 1GHz ARM11 και 512MB LPDDR2 SDRAM, διαθέτει Wifi Lan και bluetooth αλλά δεν διαθέτει ενσωματωμένες usb θύρες.



Εικόνα 17 - Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi Zero

## Περισσότερα σχετικά με το Raspberry Pi

Το Raspberry pi βασίζεται στο SoC Broadcom BCM2835. SoC σημαίνει System on Chip και είναι μία μέθοδος στην οποία τοποθετούμε όλα τα απαραίτητα ηλεκτρονικά προκειμένου να λειτουργήσει ένας υπολογιστής σε ένα μόνο chip. Έτσι αντί να έχουμε ξεχωριστά chip για την μνήμη, τον επεξεργαστή ή την gpu τα έχουμε όλα σε ένα chip. Το BCM2835 περιέχει έναν ARM1176JFZS κινητής υποδιαστολής με συχνότητα 700MHz και μία Videocore 4 GPU.

Το BCM2835 περιέχει μία πληθώρα περιφερειακών συσκευών που μπορούν να προσβαστούν από την ARM CPU. Στην ακόλουθη λίστα παρουσιάζονται αυτές οι συσκευές :

- Timers
- Interrupt controller
- GPIO
- USB
- PCM / I2S
- DMA controller
- I2C master
- I2C / SPI slave
- SPI0, SPI1, SPI2
- PWM
- UART0, UART1

Γενικότερα το BCM2835 διαθέτει 54 gpio ( General Purpose Input/Output ) lines. Πέραν της λειτουργίας τους ως ψηφιακοί είσοδοι/έξοδοι μπορούν να χρησιμοποιηθούν και σε κάποια εναλλακτική λειτουργία. Αυτή η εναλλακτική λειτουργία συνήθως αφορά κάποια περιφερειακή συσκευή, για παράδειγμα μπορεί να ρυθμιστεί να παράγει PWM (Pulse Width Modulation) παλμό. Το μέγιστο εναλλακτικών λειτουργιών για μια ακίδα gpio είναι πέντε. Όλες οι ακίδες δέχονται τάση από 0 έως 3.3 V εκτός από τις 5V ακίδες. Η μέγιστη επιτρεπτή ροή ρεύματος

από τις ακίδες των 3.3V είναι 50mA συνολικά και 16mA από την καθεμία. Αντίθετα από τις ακίδες των 5V η ροή περιορίζεται στην τροφοδοσία μείον αυτό που ρέει στο board. Το model B έχει απαιτήσεις 500mA περίπου, η τροφοδοσία παρέχει το μέγιστο 1A, άρα από τις 5V ακίδες μπορούμε να λάβουμε το πολύ 500mA . Το raspberry μπορεί να χρησιμοποιήσει και κάποιο linux-based λειτουργικό σύστημα. Το σύνηθες λειτουργικό σύστημα που χρησιμοποιεί είναι το Raspbian, το οποίο είναι ειδικά σχεδιασμένο για να χρησιμοποιηθεί με το raspberry pi. Το λειτουργικό σύστημα αποθηκεύεται στην SD Card. Πέραν από το Raspbian υπάρχουν και κάποια λειτουργικά συστήματα που μπορούν να χρησιμοποιηθούν με το raspberry. Ο install manager για το raspberry pi είναι ο NOOBS (New Out Of Box Software), και στην εκτέλεση του δίνει την δυνατότητα επιλογής ανάμεσα σε αυτά τα λειτουργικά συστήματα. Τέλος να αναφερθεί ότι μπορεί να γίνει χρήση του raspberry χωρίς κάποιο λειτουργικό σύστημα, κάνοντας χρήση κάποιου compiler για ARM επεξεργαστές και τοποθετώντας το δυαδικό αρχείο με το πρόγραμμα στην SD Card.

#### **Μερικά από τα πλεονεκτήματα του Raspberry PI είναι τα εξής:**

- Μικρό κόστος
- Μέγεθος αντίστοιχο μιας πιστωτικής κάρτας
- Χαμηλή κατανάλωση ενέργειας
- Ευκολία στην χρήση
- Αναπαραγωγή HD Video
- Δυνατότητα μετατροπής συχνότητας επεξεργαστή (Over clocking)
- Επέκταση λειτουργιών της πλακέτας μέσω μια σειρά ακίδων

#### **Μειονεκτήματα τα όποια μπορούμε να συμπεριλάβουμε:**

- Μη επεκτάσιμη μνήμη Ram
- Αργή υπολογιστική ισχύς του επεξεργαστή
- Μη λειτουργία των ενεργοβόρων συσκευών μέσω USB

### 2.1.2 USB Camera

Αν και αρχικά σκεφτόμασταν να χρησιμοποιήσουμε ένα Raspberry Pi Camera Module τελικά επιλέξαμε μία usb camera που είχαμε ήδη λόγω ευκολίας και κόστους .Η κάμερα είναι η Logitech HD Webcam C270 με τα εξής χαρακτηριστικά μέγιστη ανάλυση φωτογραφίας 3mp και ανάλυση βίντεο 1280x720 (HD ανάλυση) με μέγιστο frame rate 30fps. Η σύνδεση πραγματοποιείται μέσω usb .



Εικόνα 18 - Usb Camera Logitech Webcam C270

### 2.1.3 Micro SD Card

**Κάρτα μνήμης** ή **κάρτα φλας** (Micro SD) είναι μια συσκευή αποθήκευσης ψηφιακών δεδομένων. Χρησιμοποιείται πολύ συχνά σε συσκευές όπως ψηφιακές φωτογραφικές μηχανές, κινητά τηλέφωνα, υπολογιστές κλπ. Οι κάρτες μνήμης είναι μικρές σε μέγεθος, επανεγγράψιμες και διατηρούν τα δεδομένα τους χωρίς την ανάγκη για τροφοδοσία με ρεύμα.



Εικόνα 19 - Micro SD Sandisk Κάρτα Μνήμης

## 2.1.4 Σάσι (Chassis) με ηλεκτρικούς κινητήρες

### 2.1.4.α Σάσι (Chassis)

Ένα σασί αποτελεί το εσωτερικό πλαίσιο οχήματος το οποίο στηρίζει ένα τεχνητό αντικείμενο κατά την κατασκευή και τη χρήση του. Μπορεί επίσης να προσφέρει προστασία για ορισμένα εσωτερικά μέρη. Το δικό μας σασί περιλαμβάνει δύο επίπεδα από πλαστικό, δυο τροχούς και δύο dc κινητήρες οι οποίοι κινούν τους τροχούς.



Εικόνα 20 - Σασί

### 2.1.4.β Ηλεκτρικοί κινητήρες

Ο ηλεκτρικός κινητήρας είναι η συσκευή που τελικά αναλαμβάνει την μετατροπή της ηλεκτρικής ενέργειας σε μηχανική και αντίστροφα. Η επιλογή του είναι καθοριστικής σημασίας για την απόδοση του ηλεκτρικού συστήματος κίνησης και συνεπώς θα πρέπει να βασίζεται σε αυστηρά κριτήρια που περιλαμβάνουν την υψηλή απόδοση, υψηλή ροπή σε μικρές ταχύτητες, παροχή σταθερής ισχύς σε ευρεία περιοχή λειτουργίας, χαμηλό κόστος κατασκευής και συντήρησης και μεγάλη διάρκεια ζωής. Στα σύγχρονα ηλεκτρικά οχήματα συναντάται μεγάλη ποικιλία ηλεκτρικών κινητήρων. Οι πιο διαδεδομένες επιλογές όμως είναι:

- Ο κινητήρας συνεχούς ρεύματος: DC Motor
- Ο κινητήρας επαγωγής: Induction Motor (IM)
- Ο σύγχρονος κινητήρας μονίμων μαγνητών: Permanent Magnet Synchronous Machine (PMSM)
- Ο κινητήρας μαγνητικής αντίδρασης: Switched Reluctance Motor (SRM)

#### Κινητήρας συνεχούς ρεύματος (DC Motor)

Εμείς θα αναλύσουμε του κινητήρες dc γιατί αυτούς χρησιμοποιούμε για την κίνηση των τροχών. Οι κινητήρες συνεχούς ρεύματος είναι οι πρώτοι που χρησιμοποιήθηκαν στα ηλεκτρικά οχήματα κυρίως λόγω το ότι η τεχνολογία τους είχε ωριμάσει από πολύ νωρίς. Ο DC κινητήρας βασίζει την λειτουργία του στην αλληλεπίδραση δύο μαγνητικών πεδίων, της διέγερσης και του τυμπάνου. Το πεδίο του τυμπάνου παράγεται από το τύλιγμα που είναι κατανεμημένο στον δρομέα, ενώ το πεδίο της διέγερσης παράγεται είτε από τύλιγμα διεγέρσεως στον στάτη, είτε από μόνιμους μαγνήτες. Εξαιτίας της ορθογωνίας θέσης του πεδίου του τυμπάνου με αυτό της διέγερσης απαιτούν απλό έλεγχο, ενώ η χρήση των μονίμων μαγνητών έχει σαν αποτέλεσμα κατασκευή κινητήρων με μικρότερη διάμετρο. Η αξιοπιστία όμως των κινητήρων αυτών περιορίζεται από την ανάγκη ύπαρξης ψηκτρών στον συλλέκτη που αναλαμβάνουν την μηχανική ανόρθωση της παραγόμενης τάσης από το τύλιγμα του τυμπάνου. Συνεπώς ενώ αρχικά οι κινητήρες συνεχούς ρεύματος ήταν η βασική επιλογή για εφαρμογές μεταβλητής ταχύτητας, η χαμηλή πυκνότητα ισχύος τους σε σχέση με εναλλακτικές τεχνολογίες, το κόστος συντήρησης των ψηκτρών (περίπου κάθε 3000 ώρες λειτουργίας) και η χαμηλή τους απόδοση (αν και μπορεί να επιτευχθεί και απόδοση πάνω από 85%) οδήγησαν στην αναζήτηση διαφορετικών τοπολογιών κινητήρων για τις εφαρμογές της ηλεκτροκίνησης.



Εικόνα 21 - Dc Motors

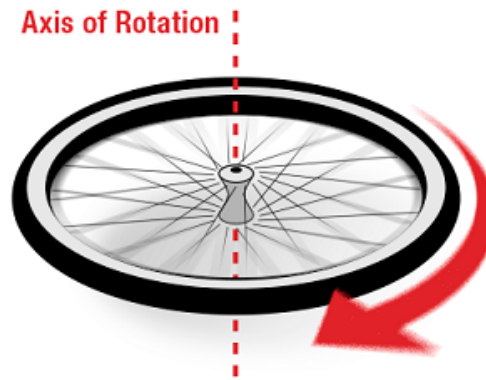
### 2.1.5 MPU6050 (Γυροσκόπιο , Επιταχυνσιόμετρο)

#### Γυροσκόπιο

Στην εφαρμογή μας χρησιμοποιούμε έναν αισθητήρα που λέγεται γυροσκόπιο. Ένα γυροσκόπιο είναι ένας τροχός που γυρίζει στον οποίο ο άξονας είναι ελεύθερος να αναλάβει οποιοδήποτε προσανατολισμό από μόνος του, επιπλέον τα γυροσκόπια μετρούν και διατηρούν περιστροφική κίνηση. Κατά την περιστροφή ο προσανατολισμός του άξονα δεν επηρεάζεται από την κλίση και η περιστροφή της τοποθέτησής του, σύμφωνα με την διατήρηση της στροφορμής. Εξαιτίας αυτού, τα γυροσκόπια είναι χρήσιμα για τη μέτρηση ή τη διατήρηση προσανατολισμού. Υπάρχουν επίσης γυροσκόπια που βασίζονται σε άλλες αρχές λειτουργίας, όπως το ηλεκτρονικό, συσκευασμένα μικροσίπ MEMS (micro-electromechanical systems) γυροσκόπια που βρέθηκαν στις ηλεκτρονικές συσκευές συσκευών, γυροσκόπια οπτικών ινών και εξαιρετικά ευαίσθητα κβαντικά γυροσκόπια. Τα γυροσκόπια μπορούν να χρησιμοποιηθούν για την κατασκευή πυξίδων , οι οποίες συμπληρώνουν ή αντικαθιστούν μαγνητικές πυξίδες (σε πλοία, αεροσκάφη και διαστημόπλοια, δηλαδή οχήματα σε γενικές γραμμές).

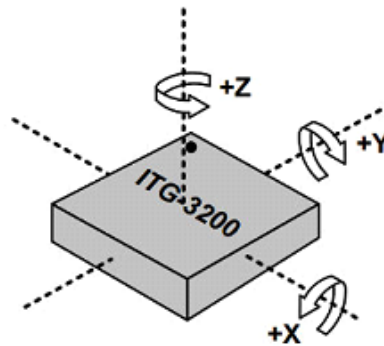
#### Πως δουλεύει ένα γυροσκόπιο

Όταν τα πράγματα περιστρέφονται γύρω από έναν άξονα που έχουν, αυτό ονομάζεται γωνιακή ταχύτητα. Ένας περιστρεφόμενος τροχός μπορεί να μετρηθεί σε στροφές ανά δευτερόλεπτο (RPS) ή μοίρες ανά δευτερόλεπτο ( o/sec).



Εικόνα 22 - Περιστρεφόμενος τροχός

Ο άξονας z που υπάρχει στο γυροσκόπιο που χρησιμοποιούμε στην ουσία είναι άξονας περιστροφής.



Εικόνα 23 - 3 Axis Digital Gyro

Ένα τριπλό γυροσκόπιο MEMS, με εκείνο που απεικονίζεται παραπάνω, μπορεί να μετρήσει την περιστροφή γύρω από τρεις άξονες: x, y και z. Μερικά γυροσκόπια έρχονται με μονούς και διπλούς άξονες τα οποία δεν είναι τόσο δημοφιλείς όσο τα γυροσκόπια με τρεις άξονες και επιπλέον είναι ενιαία σε ένα τσιπ (single chip).

Τα γυροσκόπια χρησιμοποιούνται συχνά σε αντικείμενα που δεν περιστρέφονται πολύ γρήγορα. Για παράδειγμα τα αεροπλάνα περιστρέφονται σε πολύ μικρούς βαθμούς σε κάθε άξονα. Με την ανίχνευση αυτών των μικρών αλλαγών τα γυροσκόπια βοηθούν στη σταθεροποίηση της πτήσης του αεροσκάφους.

Τα γυροσκόπια λειτουργούν με βάση δύο αρχές. Σύμφωνα με την αρχή της διατήρησης της στροφορμής, μία περιστρεφόμενη μάζα διατηρεί σταθερή την κατεύθυνση του άξονα περιστροφής. Ένα περιστρεφόμενο γυροσκόπιο κάνει ένα δορυφόρο να είναι σταθερός και πάντοτε στραμμένος προς τη Γη. Μία περιστρεφόμενη μάζα επίσης αντιστέκεται στις δυνάμεις που προσπαθούν να αλλάξουν την κατεύθυνση του άξονα περιστροφής. Στα μέσα του περασμένου



αιώνα, τεράστια γυροσκόπια, που ζύγιζαν τόνους, βιδώνονταν στην άτρακτο των πλοίων και περιστρέφονταν από κινητήρες ώστε να πιέζουν αντίθετα όταν τα κύματα έτειναν να ανατρέψουν τα σκάφη.

**Micro-Electro-Mechanical Systems, ή MEMS**, είναι μια τεχνολογία που στη γενική της μορφή μπορεί να οριστεί ως ηλεκτρονική μικρογραφία και ηλεκτρομηχανικά στοιχεία που γίνεται χρησιμοποιώντας τεχνικές μικροκατασκευής.

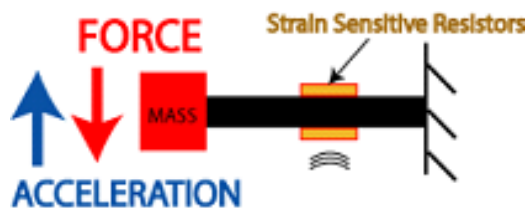
## Επιταχυνσιόμετρο

### Πως δουλεύει το Επιταχυνσιόμετρο

Το Επιταχυνσιόμετρο είναι μια ηλεκτρομηχανική συσκευή που έχει την ικανότητα να μετρά δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως είναι η επιτάχυνση της βαρύτητας, ή δυναμικές όταν προκαλούνται – προέρχονται από αλλαγές στην ταχύτητα ή στην διεύθυνση της κίνησης (επιταχύνσεις, επιβραδύνσεις, στροφές). Ένας αισθητήρας επιτάχυνσης αποτελείται από μία μάζα που συναρμολογείται πάνω σε σπειροειδή ελατήρια με τέτοιο τρόπο ώστε η μάζα να μπορεί να μετακινηθεί σε μία διεύθυνση. Αν υπάρξει επιτάχυνση  $a$  σε αυτή την διεύθυνση, η μάζα  $m$  μετακινείται κατά απόσταση  $x$ . Αυτή η αλλαγή στη θέση μπορεί να μετρηθεί μέσω διαφόρων μεθόδων και έτσι μπορούμε να έχουμε την τιμή της τρέχουσας επιτάχυνσης. Υπάρχουν τριών ειδών αισθητήρες επιτάχυνσης:

### Επιταχυνσιόμετρα πίεσης (piezoresistive accelerometer)

Τα επιταχυνσιόμετρα αυτού του τύπου, ενσωματώνουν μετρητές τάσεως οι οποίοι μετρούν την τάση που αναπτύσσεται από μια μάζα η οποία είναι προσαρτημένη σε μια ελαστική άρθρωση. Μια απλοποιημένη εκδοχή τέτοιου αισθητήρα φαίνεται στην παρακάτω εικόνα.



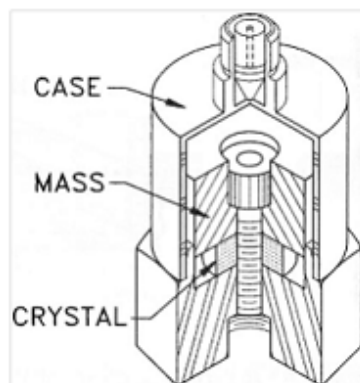
Εικόνα 24 - Επιταχυνσιόμετρο Πίεσης

Όταν επιταχύνεται αυτή η διάταξη η μάζα, λόγω αδράνειας, ασκεί μια τάση στους αισθητήρες και αυτή η τάση μετατρέπεται σε μετρούμενη επιτάχυνση.

- Πιεζοηλεκτρικά Επιταχυνσιόμετρα

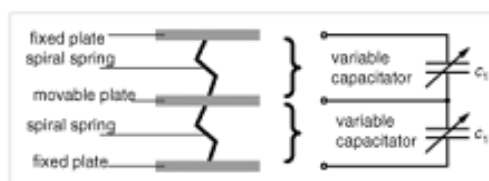
Το πιεζοηλεκτρικό φαινόμενο έχει άμεση εφαρμογή στους αισθητήρες επιτάχυνσης. Μέσω αυτού του φαινομένου υπάρχει μια άμεση μετατροπή της μηχανικής ενέργειας

σε ηλεκτρική . Αυτό επιτυγχάνεται με τη χρήση ενός κρυσταλλικού υλικού ,όπως κρυστάλλους χαλαζία οι οποίοι χρησιμοποιούνται περιστασιακά ως αισθητήρια στοιχεία, ή πιεζοηλεκτρικά κεραμικά υλικά, όπως το τιτανικό βάριο τα οποία είναι πιο δημοφιλή. Ένας τέτοιος αισθητήρας φαίνεται στην εικόνα 25. Μέσα σε μια θήκη είναι τοποθετημένος ο κρύσταλλος στριμωγμένος κάτω από μια μάζα. Όταν το σύστημα επιταχύνεται η μάζα πιέζει τον κρύσταλλο και μετρώντας τη διαφορά δυναμικού που αναπτύσσεται λόγω του πιεζοηλεκτρικού φαινομένου μπορούμε να υπολογίσουμε την επιτάχυνση που υφίσταται το σύστημα.



**Εικόνα 25 - Πιεζοηλεκτρικό Επιταχυνσιόμετρο**

- Στις περισσότερες περιπτώσεις, μεταξύ αυτών και στα «έξυπνα» κινητά, η μέτρηση γίνεται με τη χωρητική μέθοδο διότι είναι μια δοκιμασμένη και αξιόπιστη μέθοδος.



**Εικόνα 26 - Απλοποιημένο Διάγραμμα Πιεζοηλεκτρικού Επιταχυνσιόμετρου**

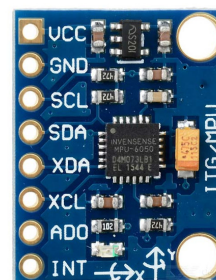
Η εικόνα 26 δείχνει ένα απλοποιημένο σχέδιο ενός αισθητήρα αυτού του είδους: Τρία φύλλα πυριτίου, τα οποία τοποθετούνται παράλληλα το ένα στο άλλο και συνδέονται με σπειροειδή ελατήρια, σχηματίζουν μια σύνδεση πυκνωτών σε σειρά. Τα δύο εξωτερικά φύλλα είναι σταθερά ενώ το εσωτερικό, το οποίο παίζει το ρόλο της μάζας, έχει τη δυνατότητα να μετακινείται. Οποιαδήποτε μετακίνηση του μεσαίου φύλλου προκαλεί διαφορά στην χωρητικότητα των πυκνωτών η οποία μετρείται και μετατρέπεται σε μετρούμενη επιτάχυνση. Για να κυριολεκτήσουμε, ως εκ τούτου δεν είναι αισθητήρες επιτάχυνσης, αλλά αισθητήρες δύναμης. Για την μέτρηση της επιτάχυνσης σε τρεις διαστάσεις θα πρέπει να έχουμε τρεις τέτοιους αισθητήρες. Αυτοί οι αισθητήρες πρέπει να είναι τοποθετημένοι κάθετα μεταξύ τους έτσι ώστε να μετρούμε την επιτάχυνση στους τρεις χωρικούς άξονες, x, y, και z-άξονας (ax, ay,

az) ανεξάρτητα. Οι αισθητήρες επιτάχυνσης χρησιμοποιούνται κατά κόρον σε διάφορες λειτουργίες των κινητών τηλεφώνων. Με αυτούς η συσκευή «καταλαβαίνει» πως είναι στραμμένη και περιστρέφει ανάλογα την οθόνη της. Σε διάφορα παιχνίδια επιτρέπεται η δυναμική αλληλεπίδραση του παίκτη μιας οι κινήσεις που ο ίδιος προκαλεί στη συσκευή αλληλεπιδρούν με το περιβάλλον του παιχνιδιού, ενώ σε κάποιες συσκευές συγκεκριμένες κινήσεις του τηλεφώνου ενεργοποιούν διάφορες λειτουργίες όπως αλλαγή σταθμού στο ραδιόφωνο, ενεργοποίηση ανοιχτής ακρόασης κτλ.

Τα επιταχυνσιόμετρα χρησιμοποιούν την αρχή του πιεζοηλεκτρικού φαινομένου για να δουλέψουν και να μας δώσουν την τιμή που θέλουμε, η οποία είναι ο ρυθμός της ταχύτητας ενός αντικειμένου Μετρούν μέτρα ανά δευτερόλεπτο (m/s) ή την δύναμη G (g). Μιας μεταβλητής η δύναμη G ισοδυναμεί με  $9,8 \text{ m/s}^2$  αλλά αυτό δεν διαφέρει με την ανύψωση (και θα είναι μια διαφορετική τιμή σε διαφορετικούς πλανήτες που οφείλεται στις διακυμάνσεις της βαρυτικής έλξης). Υπάρχουν επιταχυνσιόμετρα που μετράνε την ταχύτητα σε ένα, δύο ή τρεις άξονες, αν και να μετράνε την ταχύτητα σε 3 άξονες είναι πιο συνηθισμένο λόγω του χαμηλού κόστους. Όλα τα επιταχυνσιόμετρα περιέχουν μικροσκοπικές δομές κρυστάλλου που πιέζονται από τις δυνάμεις επιτάχυνσης, με αποτέλεσμα να παράγουν τάση, αυτό μπορεί να γίνει και μέσω της ανίχνευσης αλλαγών στη χωρητικότητα. Η αλλαγή στην χωρητικότητα γίνεται αν έχουμε δυο μικροδομές και η μια είναι δίπλα στην άλλη, οπότε έχουν μια ορισμένη χωρητικότητα μεταξύ τους, Αν μια από αυτές τις δομές κινηθεί λόγω της ταχύτητας επιτάχυνσης του οχήματος, τότε θα έχουμε αλλαγή στην χωρητικότητα. Για να πάρουμε την τιμή του επιταχυνσιόμετρου προσθέτουμε μερικά κυκλώματα για την μετατροπή της χωρητικότητας σε τάση. Τα επιταχυνσιόμετρα είναι χρήσιμα για την ανίχνευση ευαίσθητων δονήσεων σε συστήματα ή για εφαρμογές προσανατολισμού.

### IMU (Inertial Measurement Unit)

Γυροσκόπιο και Επιταχυνσιόμετρο συνδυάζονται σε μία μονάδα μέτρησης IMU (inertial measurement unit) είναι μια ηλεκτρονική συσκευή η οποία καταγράφει και αναφέρει κάποιες πληροφορίες του οχήματος, όπως η γωνία, η δύναμη που ασκείται, και το μαγνητικό πεδίο του σώματος συνδυάζοντας το επιταχυνσιόμετρο (accelometers) και το γυροσκόπιο (gyroscopes). Η λειτουργία βασίζεται στο MEMS (Micro-Electro-Mechanical Systems) Το accelerometer παράγει 3 αναλογικά σήματα περιγράφοντας την ταχύτητα και στους τρεις άξονες που κινείται το όχημα, την ίδια δουλειά κάνει το gyroscope αλλά τα σήματα περιγράφουν την γωνία που έχει το όχημα σχετικά με κάθε άξονα. Αυτά τα δεδομένα



Εικόνα 27 - MPU

συλλέγονται με το ολοκληρωμένο MPU6050 μέσω ενός ADC (Analog to Digital converter) μετατροπέα, εκτός από αυτό, η επικοινωνία του με άλλες συσκευές γίνεται μέσω του πρωτοκόλλου I2C (Inter Integrated Circuit).

### 2.1.6 Το ολοκληρωμένο L298N

Το L298N είναι μια πλακέτα με ολοκληρωμένο κύκλωμα (IC) H-bridge. Χρησιμοποιήσαμε αυτήν την πλακέτα έναντι ενός απλού ολοκληρωμένου L293D για ευκολία χώρου και δική μας πρακτικότητα. Οι οδηγοί κινητήρων λειτουργούν ως ενισχυτές ρεύματος αφού λαμβάνουν ένα σήμα ελέγχου χαμηλού ρεύματος και παρέχουν σήμα υψηλότερου ρεύματος. Αυτό το σήμα υψηλότερου ρεύματος χρησιμοποιείται για την οδήγηση των κινητήρων.

**Τα Pin της πλακέτας είναι τα εξής:**

- Out 1: Έξοδος για το Μοτέρ A
- Out 2: Έξοδος για το Μοτέρ A
- Out 3: Έξοδος για το Μοτέρ B
- Out 4: Έξοδος για το Μοτέρ B (*τυπικά είναι 5v-35v*)
- GND: Γείωση
- 5v: Είσοδος 5v (*δεν χρειάζεται αν η τροφοδοσία είναι 7v-35v*)
- EnA: Ενεργοποιεί το σήμα PWM για το ΜοτέρA
- In1: Ενεργοποιεί το Μοτέρ A
- In2: Ενεργοποιεί το Μοτέρ A
- In3: Ενεργοποιεί το Μοτέρ B
- In4: Ενεργοποιεί το Μοτέρ B
- EnB: Ενεργοποιεί το σήμα PWM για το Μοτέρ B

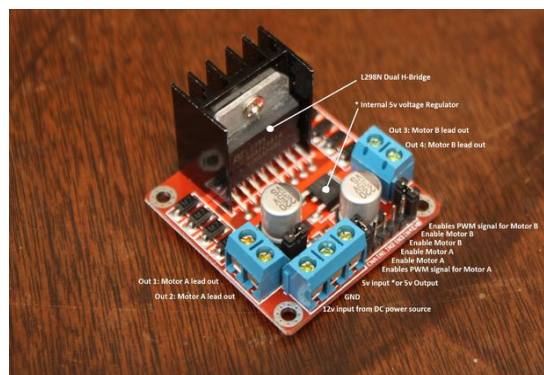
**Οι προδιαγραφές είναι οι εξής :**

- Διπλό ολοκληρωμένο οδήγησης κινητήρων H bridge: L298N
- Τάση: 5V, τάση για τους κινητήρες : 5V-35V
- Ρεύμα t: 0-36mA ρεύμα για τους κινητήρες: 2A (MAX single bridge)
- Μέγιστη ισχύ: 25W
- Διαστάσεις: 43 x 43 x 26mm
- Βάρος: 26g

Οι πίνακες αληθείας για την οδήγηση των κινητήρων παρουσιάζονται παρακάτω:

ENA	IN1	IN2	Περιγραφή
0	N/A	N/A	Motor A σβηστό
1	0	0	Motor A σταματημένο (φρένο)
1	0	1	Motor A είναι σε λειτουργία και κινείται πίσω
1	1	0	Motor B είναι σε λειτουργία και κινείται μπροστά
1	1	1	Motor A σταματημένο (φρένο)

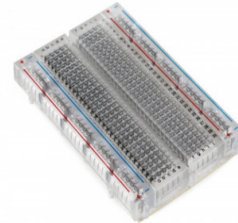
ENB	IN3	IN4	Περιγραφή
0	N/A	N/A	Motor B σβηστό
1	0	0	Motor B is stopped (brakes)
1	0	1	Motor B είναι σε λειτουργία και κινείται πίσω
1	1	0	Motor B είναι σε λειτουργία και κινείται μπροστά
1	1	1	Motor B σταματημένο (φρένο)



Εικόνα 28 - Το ολοκληρωμένο L298N

### 2.1.7 Breadboard

Είναι μία πλακέτα δοκιμών η οποία χρησιμοποιείται για ηλεκτρονικά και σχέδια κυκλωμάτων δοκιμών. Τα περισσότερα ηλεκτρονικά εξαρτήματα σε ηλεκτρονικά κυκλώματα μπορούν να διασυνδεθούν εισάγοντας τους αγωγούς ή τους ακροδέκτες τους στις οπές και στη συνέχεια πραγματοποιώντας συνδέσεις μέσω καλωδίων. Το breadboard έχει μεταλλικές ταινίες κάτω από την σανίδα και συνδέει τις τρύπες στο πάνω μέρος της σανίδας.



Εικόνα 29 - Breadboard

### 2.1.8 Θηλυκά και αρσενικά καλώδια (jumpers)

Τα jumpers είναι καλώδια που συνδέονται εύκολα είτε στο Breadboard (αρσενικά) είτε στις GPIO ακίδες του Raspberry pi.



Εικόνα 30 - Θυληκά και αρσενικά καλώδια (Jumpers)

### 2.1.9 Εξωτερικός φορτιστής μπαταριών (powerbank) και 9 Volt Μπαταρία

Εφόσον θέλουμε το όχημα μας να είναι αυτόνομο πρέπει να έχουμε μπαταρία για την τροφοδοσία του. Το power bank τροφοδοτεί το Raspberry και η 9 Volt μπαταρία τροφοδοτεί τα μοτέρ.



Εικόνα 31 - Μπαταρία 9V ,Εικόνα 32 - Powerbank

### 2.1.10 Αντάπτορας Wi-fi

Έπειτα από αρκετές δοκιμές που πραγματοποιήσαμε έχοντας κάνει σύνδεση μέσω καλωδίου ethernet στραφήκαμε στην αγορά ενός αντάπτορα wi-fi .Ο συγκεκριμένος αντάπτορας συνδέεται μέσω USB και έκανε εύκολο τον έλεγχο από απόσταση ,με αυτόν τον τρόπο έγιναν ταχύτερες και ευκολότερες οι δοκιμές μας .



Εικόνα 33 - Wi-fi Αντάπτορας

## ΚΕΦΑΛΑΙΟ 3

### 3.1 Ψηφιακή Επεξεργασία Εικόνας

#### 3.1.1 Εισαγωγή

Η ψηφιακή επεξεργασία εικόνας (Digital Image processing) είναι μια μέθοδος για τη μετατροπή μιας εικόνας σε ψηφιακή μορφή και την εκτέλεση ορισμένων λειτουργιών σε αυτήν, με αποτέλεσμα να έχουμε μια βελτιωμένη εικόνα (απαλοιφή θορύβου, καλύτερη αντίθεση) ή να εξαχθεί κάποια χρήσιμη πληροφορία από αυτήν. Είναι ένας τύπος σήματος στον οποίο η είσοδος είναι εικόνα, όπως το βίντεο ή η φωτογραφία και η έξοδος μπορεί να είναι εικόνα ή χαρακτηριστικά που σχετίζονται με αυτήν την εικόνα. Συνήθως το σύστημα επεξεργασίας εικόνων περιλαμβάνει την επεξεργασία εικόνων ως σήματα δύο διαστάσεων ενώ εφαρμόζει ήδη τις ήδη καθορισμένες μεθόδους επεξεργασίας σήματος. Είναι μεταξύ των ταχέως αναπτυσσόμενων τεχνολογιών σήμερα, με τις εφαρμογές της σε διάφορες πτυχές μιας επιχείρησης. Η Επεξεργασία Εικόνας αποτελεί τον πυρήνα της ερευνητικής περιοχής στον τομέα των μηχανικών και των επιστημών της πληροφορικής.

Η επεξεργασία εικόνας βασικά περιλαμβάνει τα ακόλουθα τρία βήματα:

- Εισαγωγή εικόνας με οπτικό σαρωτή ή με ψηφιακή φωτογραφία.
- Ανάλυση και χειρισμός της εικόνας, η οποία περιλαμβάνει τη συμπίεση δεδομένων και τη βελτίωση εικόνας και εντοπισμού που δεν είναι για τα ανθρώπινα μάτια όπως οι δορυφορικές φωτογραφίες.
- Η έξοδος είναι το τελευταίο στάδιο στο οποίο μπορεί να μεταβληθεί η εικόνα ή η έκθεση που βασίζεται στην ανάλυση εικόνας.

#### 3.1.2 Ιστορικά Στοιχεία

Παρακάτω παραθέτουμε κάποια ιστορικά στοιχεία για την εξέλιξη της ψηφιακής επεξεργασίας εικόνας.

**1921:** Μετάδοση εικόνας ηλεκτρονικά από Λονδίνο σε Νέα Υόρκη με υποθαλάσσιο καλώδιο (τηλεγράφου). Κατευθείαν εκτύπωση της εικόνας στη λήψη (όχι αποθήκευση) με τεχνική 'Half-toning'.

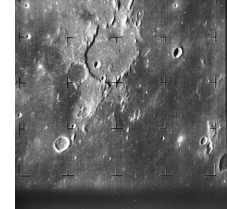
**1929:** Μετάδοση εικόνας ηλεκτρονικά από Λονδίνο σε Νέα Υόρκη και κατευθείαν εκτύπωση της με χρήση τεχνικών





φωτογραφικής αναπαραγωγής και 15 αποχρώσεις του γκρι (Βλέπε **Εικόνα 34**).

**1964:** Για πρώτη φορά χρησιμοποιούνται Ψηφιακοί Υπολογιστές για την επεξεργασία εικόνων του φεγγαριού που λήφθηκαν από φωτογραφική μηχανή στο διαστημόπλοιο Ranger 7. Ο στόχος ήταν να απαλειφθούν οι παραμορφώσεις εξαιτίας της κίνησης του διαστημοπλοίου μαζί με την φωτογραφική μηχανή (Βλέπε **Εικόνα 35**).



**1979:** Η ψηφιακή επεξεργασία εικόνας ξεκινάει να χρησιμοποιείται σε ιατρικές εφαρμογές. Ο Godfrey N. Hounsfield και ο Allan M. Cormack μοιράζονται το Βραβείο Νόμπελ στην ιατρική για την εφεύρεση της τομογραφίας. Την τεχνολογία πίσω από τον Ηλεκτρονικό αξονικό Τομογράφο.

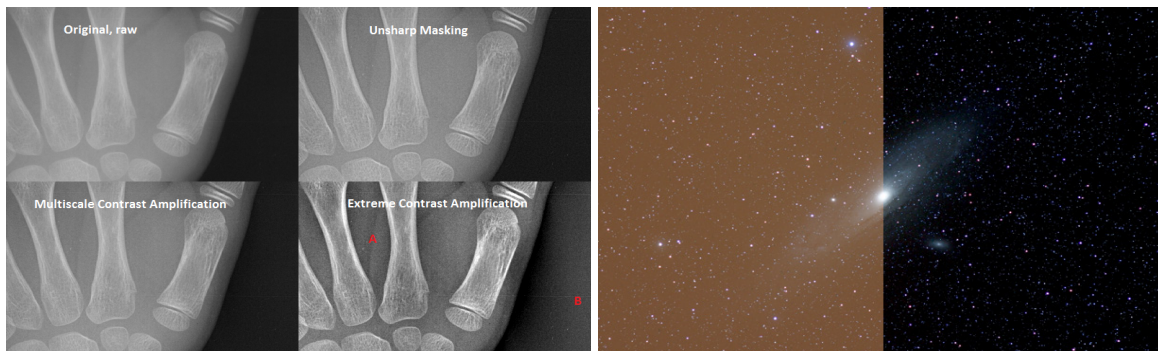
### 3.1.3 Εφαρμογές

Από το 1980 μέχρι σήμερα η Ψηφιακή επεξεργασία χρησιμοποιείται ευρέως σε διάφορες εφαρμογές που εξελίσσονται ραγδαία. Παρακάτω παραθέτουμε μερικές από αυτές:

- Βελτίωση / αποκατάσταση εικόνας
- Καλλιτεχνικά εφέ
- Ιατρική απεικόνιση
- Βιομηχανική επιθεώρηση
- Ραδιολογία
- Αστρονομία



**Εικόνα 36 - Lenna Επεξεργασία Εικόνας**



Εικόνα 37 - Διάφορες μορφές στην επεξεργασία εικόνας

Τέλος μερικές τεχνικές που χρησιμοποιούνται στην ψηφιακή επεξεργασία εικόνας περιλαμβάνουν:

- Ανάλυση κύριων τμημάτων
- Ανάλυση ανεξάρτητων τμημάτων
- Χάρτες αυτό οργάνωσης
- Κρυμμένα μοντέλα Markov
- Νευρωνικά δίκτυα

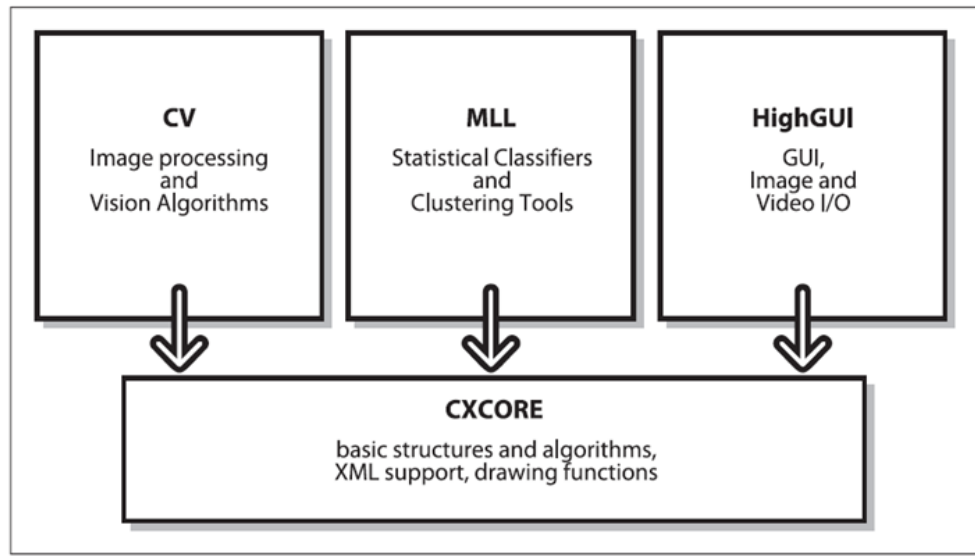
## 3.2 OpenCV και Επεξεργασία Εικόνας

### 3.2.1 Εισαγωγή στην Βιβλιοθήκη OpenCV

Η OpenCV είναι μια βιβλιοθήκη μηχανικής όρασης (computer vision) ανοικτού λογισμικού που αναπτύχθηκε αρχικά από την Intel. Η βιβλιοθήκη είναι ανεξάρτητη πλατφόρμα και παρέχει συναρτήσεις σε C, C++, Python και Java (Android). Επιπλέον διαθέτει παραπάνω από 2500 βελτιστοποιημένους αλγόριθμους. Έχει σχεδιαστεί με στόχο την αποτελεσματική εκμετάλλευση των υπολογιστικών πόρων και με έμφαση στις εφαρμογές πραγματικού χρόνου. Η OpenCV είναι διαθέσιμη σε Linux, Mac OS, Android, iOS και Windows. Η τελευταία της έκδοση είναι η 3.2.0 που κυκλοφόρησε τον Δεκέμβρη του 2016. Η πρώτη έκδοση ανακοινώθηκε το 2000 στο συνέδριο Computer Vision and Pattern Recognition και από τότε ενημερώνεται συνεχώς. Η OpenCV περιέχει συναρτήσεις κατάλληλες για πολλές διαφορετικές εφαρμογές στους τομείς της οπτικής επεξεργασίας. Διαθέτει επίσης βιβλιοθήκη μηχανικής μάθησης που περιλαμβάνει εργαλεία για ομαδοποίηση και στατιστική ταξινόμηση καθώς και συναρτήσεις για τη προβολή και την αποθήκευση των βίντεο και γενικά ό,τι έχει να κάνει με τη διεπαφή χρήστη-εφαρμογής. Τα κυριότερα πλεονεκτήματα χρήσης της OpenCV είναι η ταχύτητά της, το γεγονός ότι αποτελεί ελεύθερο λογισμικό καθώς και η απλότητα στη χρήση της.

### 3.2.2 Δομή και περιεχόμενο της OpenCV

Η OpenCV σε γενικές γραμμές διαρθρώνεται σε πέντε κύρια μέρη, τέσσερα από τα οποία παρουσιάζονται στο σχήμα. Η CV περιέχει την βασική επεξεργασία εικόνας και τους αλγόριθμους τεχνητής όρασης υψηλότερου επιπέδου. Η ML είναι η βιβλιοθήκη μάθησης μηχανής, που περιλαμβάνει πολλούς στατιστικούς ταξινομητές και εργαλεία ομαδοποίησης (clustering). Η HighGUI περιέχει ρουτίνες εισόδου/εξόδου και τις συναρτήσεις για την αποθήκευση και φόρτωση βίντεο και εικόνων και η CXCore περιέχει τις βασικές δομές δεδομένων και το περιεχόμενο. Επίσης υπάρχει η CnAux, η οποία περιέχει περιοχές που δεν χρησιμοποιούνται (ενσωματωμένο HMM αναγνώρισης προσώπου) αλλά και πειραματικούς αλγόριθμους (κατάτμηση φόντου, πρόσοψη).



Εικόνα 38 - Η δομή και το περιεχόμενο της OpenCV

### 3.2.3 Εφαρμογές της OpenCV

Παρακάτω παρουσιάζονται οι κυριότερες εφαρμογές της OpenCV:

- Αναγνώριση προσώπων/αντικειμένων
- Αναγνώριση συναισθημάτων
- Επικοινωνία ανθρώπου-υπολογιστή
- Αυτόματα συστήματα παρακολούθησης
- Συστήματα ελέγχου παραγωγής
- Συρραφή εικόνων από δορυφόρους και διαδικτυακούς χάρτες
- Ευθυγράμμιση σαρωμένων εικόνων
- Μείωση του θορύβου σε ιατρικές εικόνες
- Αντίληψη βάθους (με 2 κάμερες)
- Αναγνώριση ήχων και μουσικής, όπου οι τεχνικές αναγνώρισης εικόνων εφαρμόζονται στο φασματογράφημα του ήχου

Επιπλέον, η OpenCV αποτέλεσε ένα βασικό συστατικό του συστήματος όρασης στο ρομπότ του Stanford "Stanley" το οποίο κέρδισε βραβείο δυο εκατομμυρίων δολαρίων στο διαγωνισμό DARPA Grand Challenge Desert Robot Race .



Εικόνα 39 - Αυτόνομο Ρομπωτικό Όχημα από το Πανεπιστήμιο του Στανφορντ

### 3.2.4 Βασικές Δομές και Συναρτήσεις της OpenCV

Μερικές βασικές δομές και συναρτήσεις της OpenCV είναι τα παρακάτω:

- `IpImage`: Πίνακας στον οποίο αποθηκεύονται εικόνες. Μπορεί να έχει περισσότερα του ενός κανάλια η εικόνα.
- `CvSize`: Δομή για την αποθήκευση του μεγέθους των εικόνων.
- `CvScalar`: Δομή που έχει 4 μεταβλητές `double`, μια για κάθε κανάλι μιας εικόνας.
- `CvMat`: Πίνακας στον οποίο αποθηκεύονται στοιχεία με μία ενιαία γραμμή κώδικα.
- `Scalar`: Ο τύπος `Scalar` χρησιμοποιείται ευρέως στην OpenCV για να περάσει τιμές εικονοστοιχείων (`pixel`).
- Η συνάρτηση `cvCaptureFromCAM()` ενεργοποιεί τη κάμερα ως είσοδο ανάγνωσης εικόνων.
- Η συνάρτηση `cvSetCaptureProperty` ρυθμίζει την ανάλυση της κάμερας.
- Η συνάρτηση `cvQueryFrame` αποθηκεύει το καρτέ που δέχεται από τη κάμερα.
- Η συνάρτηση `cvReleaseCapture` ελευθερώνει τους πόρους της κάμερας.
- Η συνάρτηση `GaussianBlur` εφαρμόζει σε μία εικόνα το φίλτρο `Gaussian`.
- Η συνάρτηση `cvtColor` δέχεται ως είσοδο την αρχική εικόνα (`source image`) σε RGB και δίνει στην έξοδο την εικόνα (`destination image`) μετασχηματισμένη σε άλλο χρωματικό χώρο.

- Η συνάρτηση `moments` χρησιμοποιείται για τον υπολογισμό των ροπών μέχρι τρίτης τάξης αντικειμένου. Μόλις κληθεί η συνάρτηση αυτή όλες οι ροπές αποθηκεύονται στο διάνυσμα `m`.

### 3.2.5 Πλεονεκτήματα OpenCV

Το βασικότερο πλεονέκτημα της OpenCV είναι η ταχύτητα. Αυτός είναι και ο βασικός λόγος που δημιουργήθηκε εξ αρχής, καθώς η Intel ήθελε να δείξει πόσο προχωρημένοι ήταν οι επεξεργαστές της, που μπορούσαν να επεξεργάζονται βίντεο σε πραγματικό χρόνο. Το δεύτερο είναι ότι ενώ είναι ελεύθερο λογισμικό, υποστηρίζεται από εταιρίες όπως η Intel και η IBM, πανεπιστήμια όπως το Stanford, το MIT και πλήθος άλλων οργανισμών και εργαστηρίων που βοηθούν συνεχώς στην ανάπτυξή της. Ένα άλλο πλεονέκτημα είναι ότι δεν απαιτεί περίπλοκα μηχανήματα και εξοπλισμούς, αφού και η πιο απλή web camera, είναι αρκετή για είσοδο εικόνας και ο πιο απλός προσωπικός υπολογιστής, είναι ικανός για την επεξεργασία της.

## 3.3 Η Επεξεργασία Εικόνας στην πράξη

### 3.3.1 Η γλώσσα προγραμματισμού Python στην επεξεργασία εικόνας

Για την υλοποίηση της πτυχιακής μας επιλέξαμε την υψηλού γλώσσα προγραμματισμού Python. Δημιουργήθηκε από τον Ολλανδό Γκβίντο βαν Ρόσσουμ (Guido van Rossum) το 1990 και αρχικά ήταν γλώσσα σεναρίων που χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba, ικανή και για κλήσεις συστήματος. Στην Python δώθηκε έμφαση στην αναγνωσιμότητα του κώδικά της και στην ευκολία χρήσης της και το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ'ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Η γλώσσα χρησιμοποιεί μεταγλωττιστή (compiler) για την δημιουργία του εκτελέσιμου κώδικα και σχετίζεται με τις γλώσσες προγραμματισμού Tcl, Perl, Scheme, Java και Ruby. Άλλο ένα σημαντικό στοιχείο είναι ότι αναπτύσσεται ως ανοιχτό λογισμικό (open source) και ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την άδεια GPL.

### Οι σημαντικότεροι λόγοι που επιλέξαμε την Python είναι οι παρακάτω:

- Διότι αποτελεί scripting γλώσσα και γλώσσα προγραμματισμού. Αυτό σημαίνει ότι αν και είναι μία διερμηνευόμενη γλώσσα (δεν χρειάζεται να μεταγλωττίσετε τον κώδικά σας, απλά σώζεται και «τρέχετε» την εφαρμογή σας), διατηρεί τα περισσότερα πλεονεκτήματα μιας κλασικής γλώσσας προγραμματισμού (π.χ. Java, VB, κλπ.).
- Είναι μια «απλή γλώσσα προγραμματισμού» ενώ συχνά την κατηγοριοποιούν στις very - high level languages. Οι δύο αυτοί χαρακτηρισμοί οφείλονται κυρίως στην απλή σύνταξη και τους «γενικούς» τύπους δεδομένων (που της επιτρέπουν μεγάλο πεδίο εφαρμογής).
- Επιτρέπει τη δημιουργία modules που μπορούν να χρησιμοποιηθούν εύκολα από άλλες εφαρμογές γραμμένες σε Python.
- Είναι προσανατολισμένη στην συγγραφή μικρότερων και συμπυκνωμένων εφαρμογών. Έτσι τα προγράμματα σε Python είναι αισθητά μικρότερα από τα αντίστοιχα τους σε γλώσσες όπως οι C/C++ ή Java.
- Το όνομά της σχετίζεται με την show TOU BBC “Monty Python’s Flying Circus” και όχι με το γνωστό ερπετό.

Ο κώδικας σε γλώσσα προγραμματισμού Python μπορεί:

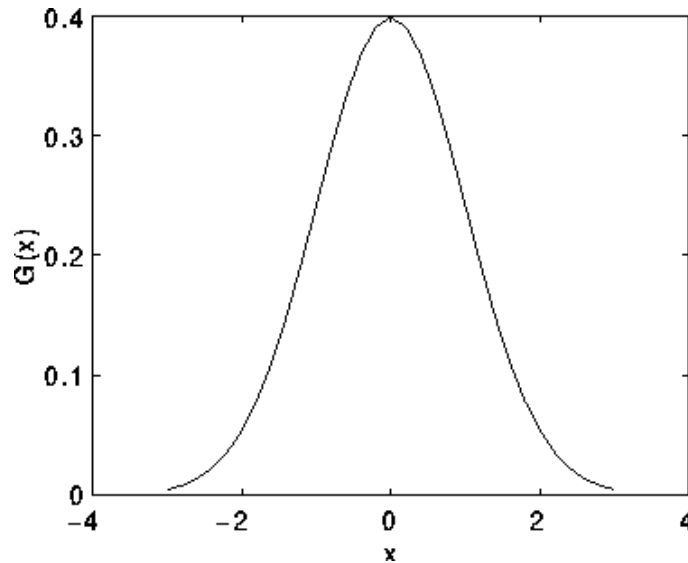
- Να χρησιμοποιηθεί σε διάφορα Λειτουργικά Συστήματα.
- Να διαβαστεί εύκολα και να τροποποιηθεί/αναπτυχθεί από άλλα άτομα, κάνοντας την ομαδική προσπάθεια πολύ πιο δυνατή και εφικτή.
- Να αναγνωριστούν εύκολα τυχόν λάθη στον κώδικα και να διορθωθούν.
- Να διευρυνθούν εύκολα οι δυνατότητες του προγράμματος με τα λεγόμενα modules (αρθρώματα) στην Python.

### 3.3.2 Φίλτρο Gaussian

Το φίλτρο Gauss αποτελεί ένα φίλτρο του οποίου η κρουστική απόκριση είναι μια συνάρτηση Gauss. Από μαθηματική άποψη, ένα φίλτρο Gauss τροποποιεί το σήμα εισόδου με συνέλιξη με τη συνάρτηση Gauss. Αυτός ο μετασχηματισμός είναι επίσης γνωστός ως μετασχηματισμός Weierstrass. Το φίλτρο Gauss είναι ένα κατωδιαβατό φίλτρο και προκαλεί την θάμπωση της εικόνας. Η κρουστική απόκριση του φίλτρου Gauss στη μία διάσταση δίνεται από την σχέση:

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

όπου  $\sigma$  είναι η τυπική απόκλιση της κατανομής .Η κατανομή φαίνεται στην παρακάτω εικόνα .

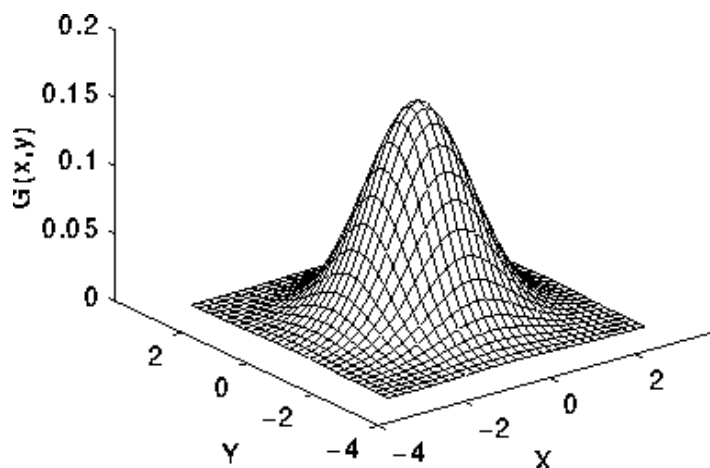


Εικόνα 40 - Κατανομή του φίλτρου Gauss σε δύο διαστάσεις

Η κρουστική απόκριση του φίλτρου Gauss στις δύο διαστάσεις είναι το γινόμενο δύο τέτοιων φίλτρων Gauss -ενός ανά διεύθυνση- και δίνεται από την σχέση:

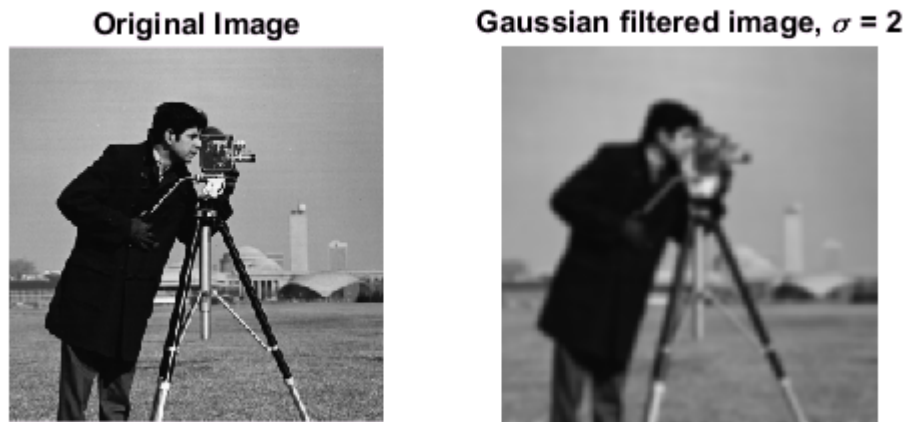
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

όπου  $x$  είναι η απόσταση από τον οριζόντιο άξονα,  $y$  είναι η απόσταση από τον κάθετο άξονα, και  $\sigma$  είναι η τυπική απόκλιση της κατανομής Gauss. Παρακάτω βλέπουμε μια κατανομή σε τρεις διαστάσεις.



Εικόνα 41 - Κατανομή του φίλτρου Gauss σε τρεις διαστάσεις

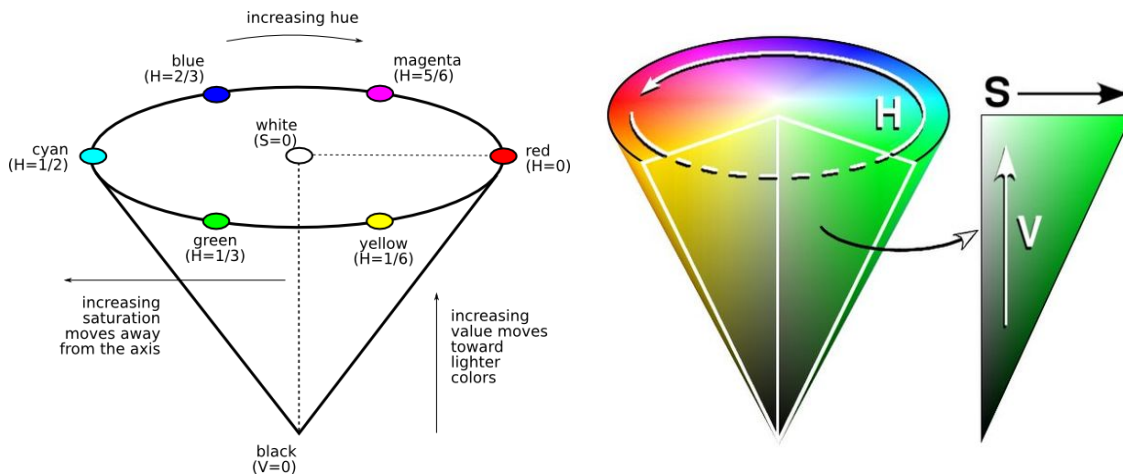




Εικόνα 42 - Εικόνα πριν και μετά τη χρήση του φίλτρου Gaussian

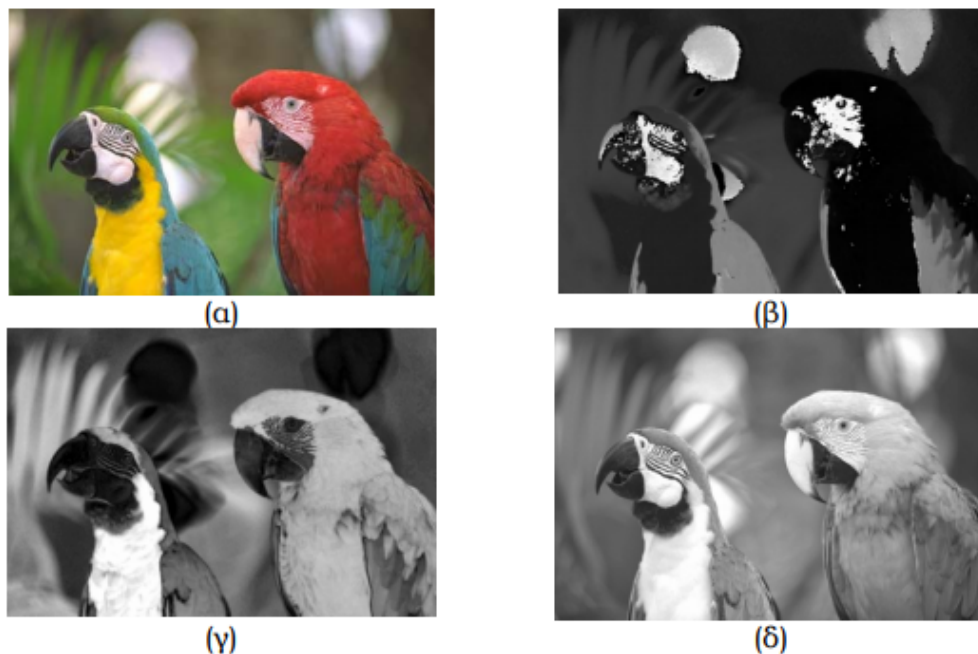
### 3.3.3 Χρωματικός χώρος HSV

Το χρωματικό μοντέλο HSV (Hue, Saturation, Value) εκμεταλλεύεται τον τρόπο που οι άνθρωποι αντιλαμβάνονται το χρώμα. Συγκεκριμένα συνηθίζεται να περιγράφουμε τις διάφορες σκηνές, όχι σε συνθήκες κόκκινου, πράσινου και μπλε, αλλά ως απόχρωση, καθαρότητα και ένταση. Βλέπουμε τα πράγματα ως χρώματα ή αποχρώσεις, οι οποίες έχουν είτε μια “ξεπλυμένη” όψη, είτε βαθύ και έντονο χαρακτήρα. Το Hue (απόχρωση) είναι το χρώμα που γίνεται αντιληπτό λόγω του μήκους κύματος. Το Saturation (καθαρότητα) είναι ο βαθμός καθαρότητας του χρώματος, δηλαδή το κατά πόσο το χρώμα έχει πρόσμιξη λευκού μέσα. Το Value (τιμή) αναφέρεται στο βαθμό μίξης ενός καθαρού χρώματος με το μαύρο. Το σύνολο των τριών αυτών ιδιοτήτων μπορεί να παράγει οποιοδήποτε χρώμα βρίσκεται στη φύση. Η τρισδιάστατη αναπαράσταση του HSV προκύπτει από τον κύβο RGB. Αν κοιτάξουμε στον RGB κύβο κατά μήκος της διαγωνίου του γκρι, μπορούμε να δούμε ένα εξάγωνο, το οποίο είναι το HSV εξάγωνο. Η απόχρωση δίνεται από τη γωνία με τον οριζόντιο άξονα με το κόκκινο στις  $0^\circ$ , το κίτρινο στις  $60^\circ$ , το πράσινο στις  $120^\circ$ , το κυανό στις  $180^\circ$ , το μπλε στις  $240^\circ$  και το ματζέντα στις  $300^\circ$ . Να σημειωθεί ότι τα συμπληρωματικά χρώματα έχουν  $180^\circ$  διαφορά. Η χρωματική καθαρότητα κυμαίνεται μεταξύ  $0.0 \leq S \leq 1.0$  και είναι ο λόγος της καθαρότητας μιας συγκεκριμένης απόχρωσης προς τη μέγιστη καθαρότητα ( $S=1$ ). Όταν  $S=0$  βρισκόμαστε στην κλίμακα του γκρι, δηλαδή στη διαγώνιο του RGB κύβου. Για την επιλογή ενός χρώματος διαλέγουμε αρχικά μια καθαρή απόχρωση (καθορίζουμε δηλαδή την τιμή του H και θέτουμε  $S=V=1$ ). Στη συνέχεια προσθέτοντας μαύρο μειώνουμε την τιμή του V και προσθέτοντας άσπρο μειώνουμε το S.



Εικόνα 43 - Χρωματικός χώρος HSV

Ο μετασχηματισμός από RGB σε HSV είναι μη γραμμικός αλλά αντιστρέψιμος. Το HSV παρουσιάζει δυο βασικά πλεονεκτήματα. Πρώτον η τιμή V είναι ανεξάρτητη από το χρώμα και δεύτερον η απόχρωση H και η χρωματική καθαρότητα S είναι στενά συσχετισμένες με τον τρόπο αντίληψης του χρώματος από το ανθρώπινο μάτι. Αυτά τα χαρακτηριστικά καθιστούν το μοντέλο HSV ιδανικό εργαλείο για την ανάπτυξη αλγορίθμων επεξεργασίας εικόνας βασισμένων στην αίσθηση χρώματος από το ανθρώπινο οπτικό σύστημα.

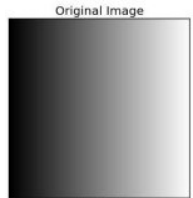

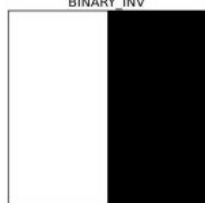


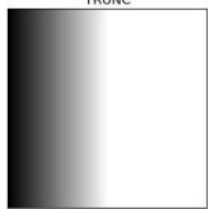
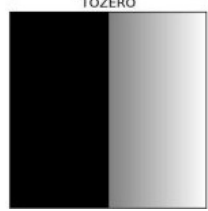

Εικόνα 44 - Παράδειγμα επεξεργασίας HSV

### 3.3.4 Threshold

Το Thresholding είναι η απλούστερη μέθοδος τμηματοποίησης εικόνας. Από μια εικόνα σε κλίμακα του γκριζου, μπορούν να χρησιμοποιηθούν κατώτατα όρια για τη δημιουργία δυαδικών εικόνων. Οι πιο απλές μέθοδοι κατωφλίου αντικαθιστούν κάθε εικονοστοιχείο σε μια εικόνα με ένα μαύρο εικονοστοιχείο εάν η ένταση εικόνας είναι μικρότερη από κάποια σταθερή σταθερή  $T$  ή ένα λευκό εικονοστοιχείο εάν η ένταση της εικόνας είναι μεγαλύτερη από αυτή της σταθεράς. Στην εικόνα παραδειγμάτων στα δεξιά, αυτό έχει ως αποτέλεσμα το σκοτεινό δέντρο να γίνει εντελώς μαύρο και το λευκό χιόνι να γίνει εντελώς λευκό.

Στο OpenCV η συνάρτηση που χρησιμοποιείται είναι `cv2.threshold`. Το πρώτο επιχείρημα είναι η εικόνα πηγής, η οποία πρέπει να είναι μια εικόνα σε αποχρώσεις του γκρι( grayscale). Το δεύτερο επιχείρημα είναι η τιμή κατωφλίου που χρησιμοποιείται για την ταξινόμηση των τιμών των εικονοστοιχείων. Το τρίτο επιχείρημα είναι το `maxVal` που αντιπροσωπεύει την τιμή που πρέπει να δοθεί εάν η τιμή των εικονοστοιχείων είναι μεγαλύτερη από (μερικές φορές μικρότερη από) την τιμή κατωφλίου. Το OpenCV παρέχει διαφορετικούς τύπους κατώτατων ορίων και αποφασίζεται από την τέταρτη παράμετρο της λειτουργίας. Διαφορετικοί τύποι είναι:

Original image	 <p>Original Image</p>
cv2.THRESH_BINARY	 <p>BINARY</p>
cv2.THRESH_BINARY_INV	 <p>BINARY_INV</p>

cv2.THRESH_TRUNC	
cv2.THRESH_TOZERO	
cv2.THRESH_TOZERO_INV	

Εικόνα 45 - Τύποι Threshold στο OpenCV

Η σύνταξη της συνάρτησης είναι :

```
ret,thresh = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
```

Η συνάρτηση μας βοήθησε να προσδιορίσουμε την μαύρη γραμμή που ακολουθεί το αυτοκινούμενο ρομπότ μας .

### 3.3.5 Εύρεση γραμμής

Παρακάτω θα αναλύσουμε την διαδικασία εύρεση γραμμής (Line Detect).Ας χωρίσουμε τα βήματα λοιπόν της διαδικασίας για την εύρεση της μαύρης γραμμής:

- Ξεκινάει η κάμερα να τραβάει βίντεο
- Διαβάζουμε την εικόνα καρέ-καρέ
- Κάνουμε την εικόνα Grayscale
- Καλούμε την συνάρτηση Threshold
- Καλούμε την συνάρτηση Gaussian
- Βρίσκουμε τα όρια της γραμμής
- Βρίσκουμε το μέσο της μαύρης γραμμής
- Συγκρίνουμε τις συντεταγμένες για να στρίψει κατάλληλα το όχημα

Ο αλγόριθμος προϋποθέτει ότι η γραμμή είναι μαύρη. Το αρχείο είναι το **line\_detect.py** και θα αναλύσουμε κάποια βασικά κομμάτια του κώδικα.

Στην παρακάτω εντολή θα ξεκινήσει η ζωντανή λήψη.

```
cap=cv2.VideoCapture(-1)
```

Έπειτα διαβάζουμε την εικόνα του βίντεο καρέ-καρέ και αφού ορίσουμε το πλάτος και το μήκος του frame, με τις συναρτήσεις Grayscale και Threshold κάνουμε την εικόνα ασπρόμαυρη και μετά χρησιμοποιούμε την συνάρτηση Gaussian για να περιορίσουμε τον θόρυβο .

```
ret, frame = cap.read()
crop = frame[180:320, 0:638]
crop2=cv2.cvtColor(crop,cv2.COLOR_BGR2GRAY)
th,crop2 = cv2.threshold(crop2,0,255,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
previous = cv2.GaussianBlur(crop2, (5,5),0)
```

Στο επόμενο βήμα με τις κατάλληλες εντολές θα βρούμε τα όρια της γραμμής δηλαδή θα προσδιορίσουμε τον χώρο που βρίσκεται η μαύρη γραμμή.

```
for cnt in contours:
    moments = cv2.moments(cnt)
    if moments['m00']!=0:
        cx = int(moments['m10']/moments['m00'])
        cy = int(moments['m01']/moments['m00'])
        moment_area = moments['m00']
        contour_area = cv2.contourArea(cnt)
        perimeter = cv2.arcLength(cnt,True)
        hull = cv2.convexHull(cnt)
        k = cv2.isContourConvex(cnt)
        x,y,w,h = cv2.boundingRect(cnt)
        rows,cols = previous.shape[:2]
        cv2.drawContours(previous, [cnt], 0, (0,255,0), 3)
        px = previous[cy,cx]
        if px == 255 :
            i=i+1
            cv2.circle(previous,(cx,cy),5,(0,0,255),-1)
```

Τέλος κάνουμε τον κατάλληλο έλεγχο για να δούμε πότε θα στρίψει το ρομπότ. Αυτό πραγματοποιείται μέσω της παρακάτω if.

```
if cx < 212 :
    print 'left'
    left()
elif cx >= 212 and cx <= 426:
    print 'center'
    center()
elif cx > 426:
    print 'right'
    right()
```

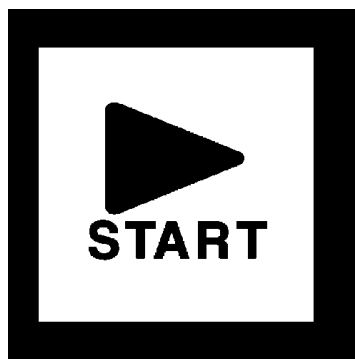
### 3.3.6 Αναγνώριση σημάτων

Παρακάτω θα αναλύσουμε την διαδικασία αναγνώρισης σημάτων. Ας χωρίσουμε τα βήματα λοιπόν της διαδικασίας για την αναγνώριση σημάτων:

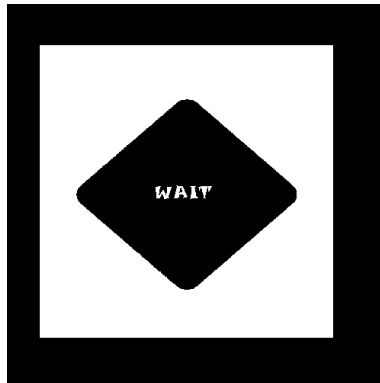
- Δημιουργήσαμε 3 εικόνες με κατάληξη .bmp (Start,Wait,Stop)
- Διαβάζουμε τις 3 εικόνες
- Διαβάζουμε την εικόνα καρέ-καρέ από την κάμερα
- Βρίσκουμε τα άκρα και το περίγραμμα
- Ανιχνεύουμε για τετράγωνα
- Συγκρίνουμε τις εικόνες με το περιεχόμενο του τετραγώνου που έχει ανιχνευτεί από την καρέ-καρέ εικόνα της κάμερας

Πρώτα δημιουργήσαμε 3 διαφορετικές εικόνες έτσι ώστε όταν τις εντοπίζει το ρομπότ να συμπεριφέρεται ανάλογα. Δημιουργήσαμε μία εικόνα για να ξεκινάει την διαδρομή και την παρακολουθήση της μαύρης γραμμής, μία για να περιμένει 5 δευτερόλεπτα και μία για να σταματάει τελείως το ρομπότ. Οι εικόνες είναι οι εξής:

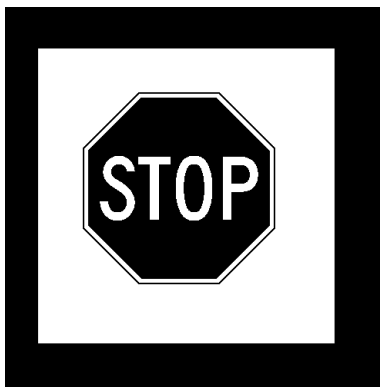
Για την εκκίνηση:



Για την αναμονή:



Για τον τερματισμό:



Στην παρακάτω συνάρτηση διαβάζουμε τις 3 εικόνες.

```
def readRefImages():  
    for count in range(3):  
        image = cv2.imread(ReferenceImages[count], cv2.COLOR_BGR2GRAY)  
        symbol[count].img = cv2.resize(image,(w/2,h/2),interpolation = cv2.INTER_AREA)  
        symbol[count].name = ReferenceTitles[count]
```

Έπειτα διαβάζουμε καρέ-καρέ την εικόνα από την κάμερα, χρησιμοποιούμε την συνάρτηση grayscale και την gaussian για να μειώσουμε τον θόρυβο.

```
ret, frames = cap.read()  
grays = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)  
blurred = cv2.GaussianBlur(grays,(3,3),0)
```

Μετά με τις παρακάτω εντολές ανιχνεύουμε τα άκρα και τις γωνίες.

```
def auto_canny(image, sigma=0.33):
    # compute the median of the single channel pixel intensities
    v = np.median(image)

    # apply automatic Canny edge detection using the computed median
    lower = int(max(0, (1.0 - sigma) * v))
    upper = int(min(255, (1.0 + sigma) * v))
    edged = cv2.Canny(image, lower, upper)

    # return the edged image
    return edged
```

Δημιουργούμε περιγράμματα και αναζητάμε για τυχόν τετράγωνα.

```
for cnt12 in contours:
    approxs = cv2.approxPolyDP(cnt12,0.01*cv2.arcLength(cnt12,True),True)
    if len(approxs)==4:
        areas = cv2.contourArea(approxs)
```

Συγκρίνουμε την περιοχή του τετραγώνου που έχει ανιχνευτεί με τις εικόνες που έχουμε δημιουργήσει.

```
for i in range(3):
    diffImgs = cv2.bitwise_xor(warped_eqs, symbol[i].img)
    diffs = cv2.countNonZero(diffImgs);
```

Τέλος ελέγχουμε εάν πρόκειται για το σήμα του Start, του Wait ή του Stop.

```
if diffs < 4500 and diffs > 3000
    print "WAIT"
elif diffs < 6200 and diffs > 4500
    print "START"
elif diffs > 8000 and diffs < 9999
    print "STOP"
```



## 3.4 Ισορροπία Οχήματος

### 3.4.1 Πως επιτυγχάνεται η ισορροπία

Τα βασικά πίσω από ένα ρομπότ εξισορρόπησης βασίζονται στην ιδέα του Inverted Pendulum . Αυτό θα επιτευχθεί με έναν αλγόριθμο ελέγχου που ονομάζεται PID(Proportional–Integral–Derivative). Το ρομπότ για να ισορροπήσει πρέπει να διατηρήσει τους τροχούς κάτω από το κέντρο βάρους. Π.χ. Εάν το ρομπότ κλίνει προς τα εμπρός, οι τροχοί γυρίζουν προς τα εμπρός προσπαθώντας να διορθώσει την κλίση.

Για να μετρήσουμε τις γωνίες χρησιμοποιούμε την πλακέτα MPU6050. Οι αισθητήρες IMU αποτελούνται συνήθως από δύο ή περισσότερα μέρη. Ταξινομώντας τα κατά προτεραιότητα, είναι το επιταχυνσιόμετρο, το γυροσκόπιο, το μαγνητόμετρο και ο υψομετρητής. Το MPU 6050 είναι ένας αισθητήρας 6 DOF (βαθμοί ελευθερίας) ή έξι αξόνων IMU, που σημαίνει ότι δίνει έξι τιμές ως έξοδο. Τρεις τιμές από το επιταχυνσιόμετρο και τρεις από το γυροσκόπιο. Το MPU 6050 είναι ένας αισθητήρας που βασίζεται στην τεχνολογία MEMS (Micro Electro Mechanical Systems). Τόσο το επιταχυνσιόμετρο όσο και το γυροσκόπιο είναι ενσωματωμένα σε ένα ενιαίο τσιπ. Αυτό το τσιπ χρησιμοποιεί το πρωτόκολλο I2C (Inter-Integrated Circuit) για επικοινωνία.

Ένα **επιταχυνσιόμετρο** λειτουργούν με βάση την αρχή του πιεζοηλεκτρικού φαινομένου. Εδώ, φανταστείτε ένα κυβοειδές κουτί με μια μικρή σφαίρα μέσα του, όπως στην παραπάνω εικόνα. Τα τοιχώματα αυτού του κουτιού κατασκευάζονται με πιεζοηλεκτρικούς κρυστάλλους. Κάθε φορά που κλίνετε το κουτί, η μπάλα αναγκάζεται να κινηθεί προς την κατεύθυνση της κλίσης, λόγω της βαρύτητας. Ο τοίχος που συγκρούεται με την μπάλα δημιουργεί μικροσκοπικά πιεζοηλεκτρικά ρεύματα. Υπάρχουν τρία ζεύγη αντίθετων τοίχων σε ένα κύβο. Κάθε ζεύγος αντιστοιχεί σε άξονα σε τρισδιάστατο χώρο: X, Y και Z άξονες. Ανάλογα με το ρεύμα που παράγεται από τα πιεζοηλεκτρικά τοιχώματα, μπορούμε να καθορίσουμε την κατεύθυνση της κλίσης και το μέγεθος της.

Τα **γυροσκόπια** λειτουργούν με βάση την αρχή της επιτάχυνσης του Coriolis. Φανταστείτε ότι υπάρχει μια δομή που μοιάζει με πιρούνι και είναι σε συνεχή κίνηση προς τα εμπρός και πίσω. Διατηρείται στη θέση του χρησιμοποιώντας πιεζοηλεκτρικούς κρυστάλλους. Κάθε φορά που προσπαθείτε να κλίσετε αυτή τη διάταξη, οι κρύσταλλοι βιώνουν μια δύναμη προς την κατεύθυνση της κλίσης. Αυτό προκαλείται ως αποτέλεσμα της αδράνειας του κινούμενου πιρουνιού. Έτσι, οι

κρύσταλλοι παράγουν ένα ρεύμα συναίνεσης με το πιεζοηλεκτρικό φαινόμενο και αυτό το ρεύμα ενισχύεται. Οι τιμές εκλέγονται από τον μικροελεγκτή υποδοχής.

### 3.4.2 Ανάγνωση των δεδομένων από το MPU6050

Για το γυροσκόπιο, πρέπει να υπολογίσουμε πόσο γρήγορα περιστρέφεται σε βαθμούς ανά δευτερόλεπτο (dps). Στη συνέχεια, πρέπει να παρακολουθήσουμε την απόκλιση γωνίας από το σημείο μηδέν. Αυτό θα το υπολογίσουμε με τους παρακάτω τύπους.

#### Ρυθμός Περιστροφής Γυροσκοπίου

```
rate_gyr_x = (float) gyr_x_raw * G_GAIN
```

**Gyr\_x\_raw** = ακατέργαστα δεδομένα που αποκτήθηκαν από τον γυροσκοπικό άξονα X.

**G\_GAIN** = 0.00875, η οποία βασίζεται στο επίπεδο ευαισθησίας που χρησιμοποιείται από το γυροσκόπιο.

**Rate\_gyr\_x** = ο ρυθμός περιστροφής ανά δευτερόλεπτο.

#### Γωνία Γυροσκοπίου

```
gyroXangle+=rate_gyr_x * DT
```

**DT** = περίοδος βρόχου.

**GyroXangle** = είναι η τρέχουσα γωνία X που υπολογίζεται από τα δεδομένα gyro X.

Έχω DT set top 0.02, η οποία είναι 20ms. Αυτός είναι ο χρόνος που χρειάζεται για να ολοκληρωθεί ένας κύκλος του κύριου βρόχου. Αυτή η περίοδος βρόχου πρέπει να είναι σταθερή και ακριβής, διαφορετικά το γυροσκόπιο σας θα παρασύρεται.

#### Γωνία Επιταχυνσιομέτρου

Η γωνία X μπορεί να υπολογιστεί χρησιμοποιώντας τριγωνομετρία και τις πρώτες τιμές από τον άξονα επιταχυνσιόμετρο Y και Z. Αυτό γίνεται χρησιμοποιώντας τη συνάρτηση [Atan2](#) για να επιστρέψετε την κύρια τιμή της εφαπτομένης των Y και Z, εκπεφρασμένη σε ακτίνια.

Προσθέτουμε π στα ακτινοειδή ώστε να έχουμε ένα αποτέλεσμα μεταξύ 0 και 2. Στη συνέχεια μετατρέπουμε τα ακτινικά σε μοίρες πολλαπλασιάζοντας τα ακτινικά κατά 57,29578 (180 / π).

```
AccXangle = (πλωτήρας) (atan2 (acc_y_raw, acc_z_raw) + M_PI) *
RAD_TO_DEG
```

**M\_PI** = 3.14159265358979323846

**RAD\_TO\_DEG** = 57.29578, 1 ακτίνα = 57.29578 μοίρες.

### Συνδυασμός Γωνιών

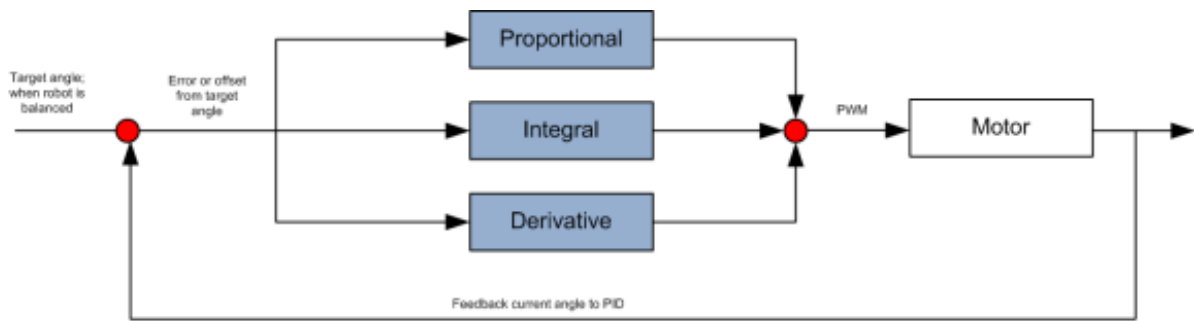
```
CFangleX = AA * (CFangleX + rate_gyr_x * DT) + (1 - AA) * AccXangle;
```

**AA** = 0,98 Συμπληρωματική σταθερά φίλτρου

Το CFangleX είναι η τελική μας γωνία που θα χρησιμοποιηθεί για την εξισορρόπηση του ρομπότ.

### 3.4.3 Ο αλγόριθμος PID (Proportional Integral Derivative)

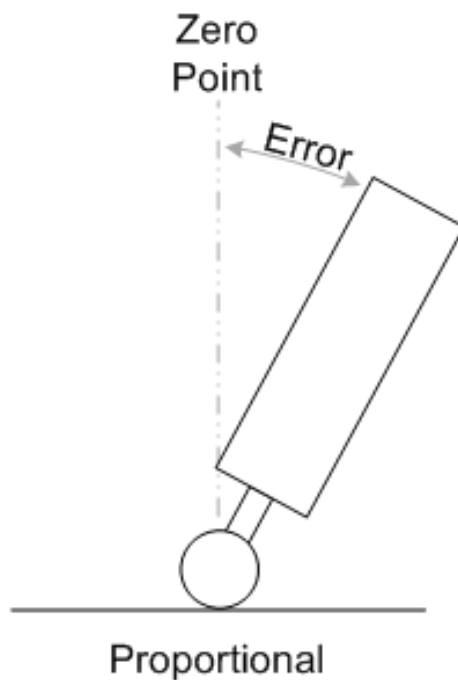
Ένας αλγόριθμος ελέγχου PID χρησιμοποιείται για την εξισορρόπηση του ρομπότ με την κίνηση των κινητήρων με βάση την κλίση.



Εικόνα 46 - Διάγραμμα ελέγχου P.I.D

Ο αναλογικός (P) όρος του PID βασίζεται στην τρέχουσα διαφορά γωνίας ή στο σφάλμα από το Point Zero (όπου τα υπόλοιπα ρομπότ) πολλαπλασιάζονται με το κέρδος P. Το κέρδος P είναι ένας αριθμός που πρέπει να ελέγχουμε για να πάρουμε το σωστό ποσό του αναλογικού ελέγχου.

Ο αναλογικός έλεγχος θα προσπαθήσει να διορθώσει την ισορροπία με βάση το τρέχον σφάλμα από το σημείο μηδέν.

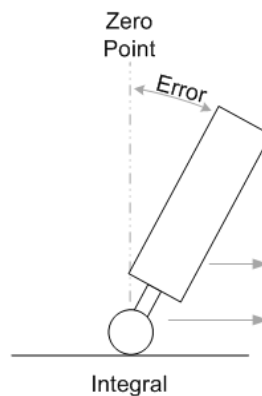


Εικόνα 47 - Αναπαράσταση της ισορροπίας

Το τμήμα του κώδικα για τον όρο P είναι:

$$P_{term} = K_P * CFangleX;$$

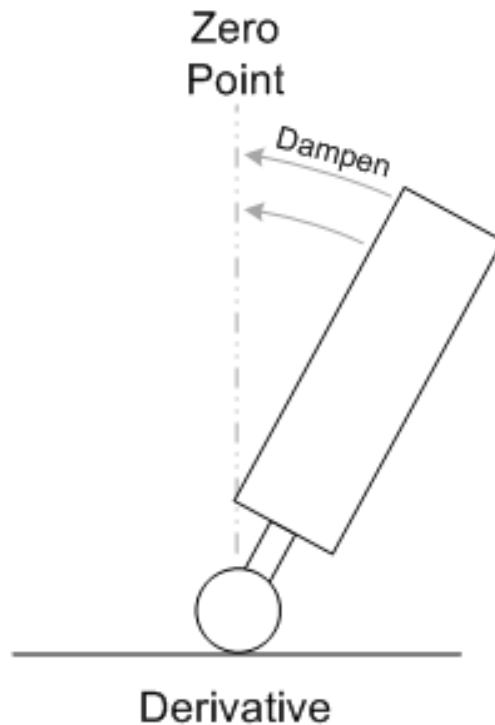
Ο ακέραιος (I) όρος του PID βασίζεται στην τρέχουσα διαφορά γωνίας ή σφάλματος από το σημείο μηδέν πολλαπλασιαζόμενο με το κέρδος I, το οποίο είναι αυτό που συσσωρεύεται με την πάροδο του χρόνου. Ο ολοκληρωμένος έλεγχος συμβάλλει στην εξισορρόπηση του ρομπότ αν κινούνται.



**Το τμήμα του κώδικα για τον όρο I είναι:**

$$ITerm + = K_I * CFangleX;$$

Ο παράγωγος (D) όρος του PID βασίζεται στον τρέχοντα ρυθμό περιστροφής. Χρησιμοποιείται για να μειώσει την απόκριση καθώς το ρομπότ φτάνει στο μηδενικό σημείο. Χωρίς το παράγωγο έλεγχο, το ρομπότ θα υπερβεί και θα αρχίσει να ταλαντεύεται.



Το τμήμα του κώδικα για τον όρο D είναι:

```
DTerm = KD * (CFangleX - lastAngle).
```

```
LastAngle = CFangleX;
```

Η προσθήκη όλων των τιμών PID θα χρησιμοποιηθεί για να οδηγήσει τους κινητήρες προς τη σωστή κατεύθυνση και με τη σωστή ισχύ.

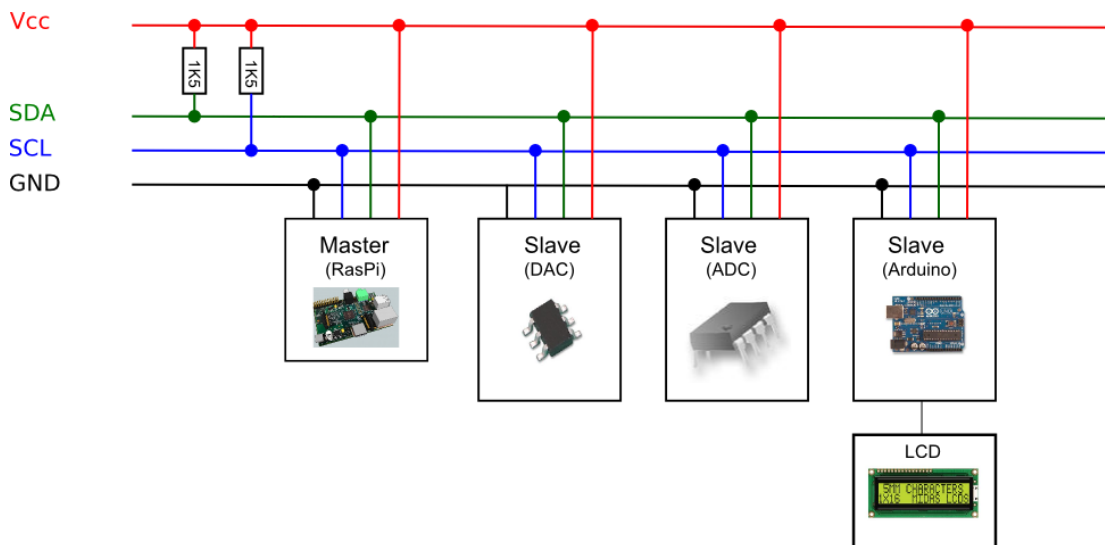
```
Έξοδος = Pterm + iTerm + dTerm;
```

### 3.4.4 Ο διάυλος επικοινωνίας I2C

Ο διάυλος επικοινωνίας I2C είναι πολύ δημοφιλής και χρησιμοποιείται ευρέως από πολλές ηλεκτρονικές συσκευές, διότι μπορεί εύκολα να εφαρμοστεί σε πολλά ηλεκτρονικά σχέδια τα οποία απαιτούν επικοινωνία μεταξύ μιας κύριας και πολλαπλών slave συσκευών. Οι εύκολες υλοποιήσεις έρχονται με το γεγονός ότι απαιτούνται μόνο δύο καλώδια για επικοινωνία μεταξύ μέχρι 128 (112) συσκευών

όταν χρησιμοποιούνται 7 bits διευθύνσεων και μέχρι σχεδόν 1024 (1008) συσκευές όταν χρησιμοποιούνται 10 bits διευθύνσεων.

Πώς είναι δυνατόν, μια επικοινωνία μεταξύ τόσων συσκευών μόνο με καλώδια; Καλά κάθε συσκευή έχει ένα προκαθορισμένο αναγνωριστικό ή μια μοναδική διεύθυνση συσκευής, έτσι ώστε ο κύριος μπορεί να επιλέξει με ποιες συσκευές θα επικοινωνούν. Τα δύο καλώδια ή γραμμές ονομάζονται σειριακό ρολόι (ή SCL) και σειριακά δεδομένα (ή SDA). Η γραμμή SCL είναι το σήμα ρολογιού που συγχρονίζει τη μεταφορά δεδομένων μεταξύ των συσκευών στο δίαυλο I2C και παράγεται από την κύρια συσκευή. Η άλλη γραμμή είναι η γραμμή SDA που μεταφέρει τα δεδομένα. Οι δύο γραμμές είναι "open-drain", πράγμα που σημαίνει ότι οι αντιστάσεις έλξης πρέπει να είναι προσαρτημένες σε αυτές έτσι ώστε οι γραμμές να είναι υψηλές, επειδή οι συσκευές στο δίαυλο I2C είναι ενεργές χαμηλές. Συνήθως χρησιμοποιούμενες τιμές για τις αντιστάσεις είναι από 2K για υψηλότερες ταχύτητες στα 400 kbps περίπου, έως 10K για χαμηλότερη ταχύτητα στα 100 kbps περίπου. Ένα παράδειγμα παρουσιάζεται στη εικόνα παρακάτω .

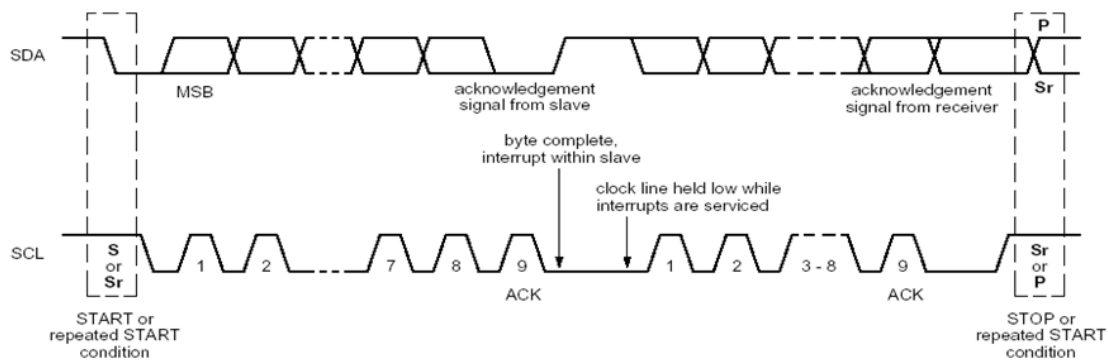


**Εικόνα 48 - Παράδειγμα επικοινωνίας I2C**

Το σήμα δεδομένων μεταφέρεται σε ακολουθίες 8 bit. Έτσι, μετά από μια ξεχωριστή κατάσταση εκκίνησης, έρχεται η πρώτη ακολουθία των 8 δυαδικών ψηφίων που υποδεικνύει τη διεύθυνση του σκλάβου στον οποίο αποστέλλονται τα δεδομένα. Μετά από κάθε ακολουθία 8 bits ακολουθεί ένα κομμάτι που ονομάζεται Acknowledge. Μετά το πρώτο bit αναγνώρισης στις περισσότερες περιπτώσεις έρχεται μια άλλη ακολουθία διευθύνσεων, αλλά αυτή τη φορά για τους εσωτερικούς καταχωρητές της εξαρτημένης συσκευής. Αφού οι ακολουθίες διευθύνσεων

ακολουθούν τις ακολουθίες δεδομένων, μέχρις ότου τα δεδομένα αποστέλλονται εντελώς και τελειώνουν με μια ειδική κατάσταση διακοπής.

Ας δούμε ακόμη πιο προσεκτικά αυτά τα γεγονότα. Η συνθήκη εκκίνησης εμφανίζεται όταν η γραμμή δεδομένων πέφτει χαμηλή ενώ η γραμμή του ωρολογίου εξακολουθεί να είναι υψηλή. Μετά από αυτό ξεκινάει το ρολόι και κάθε bit δεδομένων μεταφέρεται κατά τη διάρκεια κάθε παλμού ρολογιού. Η ακολουθία διευθύνσεων της συσκευής ασκεί πρώτα το σημαντικότερο bit (MSB) και τελειώνει με το λιγότερο σημαντικό bit (LSB) και στην πραγματικότητα αποτελείται από 7 bits επειδή το 8ο bit χρησιμοποιείται για να υποδείξει εάν ο κύριος θα γράψει στο slave (Logic low) ή θα διαβάσει από αυτό (Logic high).



**Εικόνα 49 - Παράδειγμα μεταφοράς δεδομένων**

Το επόμενο bit ACK / NACK χρησιμοποιείται από τη δευτερεύουσα συσκευή για να υποδείξει αν έχει λάβει επιτυχώς την προηγούμενη ακολουθία δυαδικών ψηφίων. Επομένως, αυτή τη στιγμή η κύρια συσκευή παραδίδει τον έλεγχο της γραμμής SDA στη εξαρτημένη συσκευή και αν η υποτελής συσκευή έχει λάβει επιτυχώς την προηγούμενη ακολουθία, θα τραβήξει τη γραμμή SDA προς την κατάσταση που ονομάζεται Acknowledge. Εάν ο σκλάβος δεν τραβήξει τη γραμμή SDA προς τα κάτω, η κατάσταση ονομάζεται Μη αναγνώριση και σημαίνει ότι δεν έλαβε επιτυχώς την προηγούμενη ακολουθία που μπορεί να προκληθεί από διάφορους λόγους. Για παράδειγμα, ο σκλάβος μπορεί να είναι απασχολημένος, μπορεί να μην καταλάβει τα ληφθέντα δεδομένα ή εντολές, δεν μπορεί να λάβει περισσότερα δεδομένα κ.ο.κ. Σε μια τέτοια περίπτωση, η κύρια συσκευή αποφασίζει πώς θα συνεχιστεί.

Ακολουθούν οι εσωτερικοί καταχωρητές που απευθύνονται. Οι εσωτερικοί καταχωρητές είναι τοποθεσίες στη μνήμη του slave που περιέχουν διάφορες πληροφορίες ή δεδομένα. Στην δική μας περίπτωση master αποτελεί το raspberry pi και slave το MPU6050, θα αναλύσουμε σε παρακάτω κεφάλαιο την σύνδεση του και την εγκατάσταση των drivers.





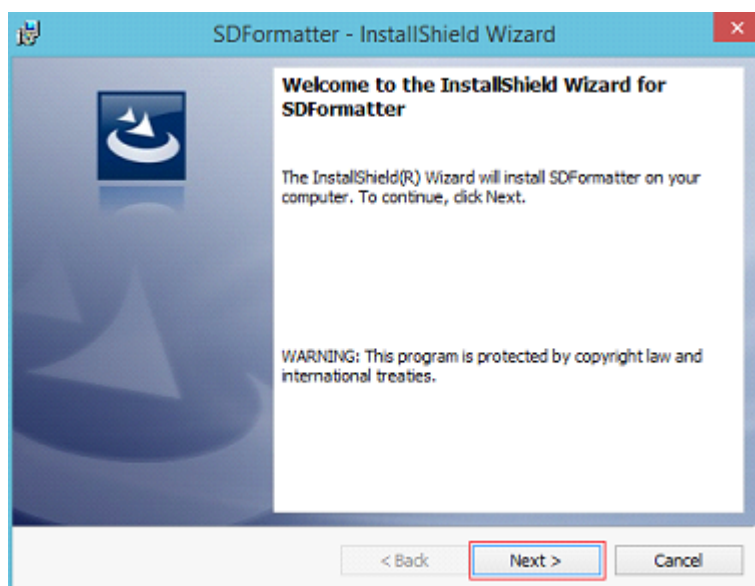
## ΚΕΦΑΛΑΙΟ 4

### 4.1 Υλοποίηση - Συναρμολόγηση κατασκευής ,Εγκατάσταση προγραμμάτων.

#### 4.1.1 Εγκατάσταση Λειτουργικού Συστήματος στο Raspberry Pi

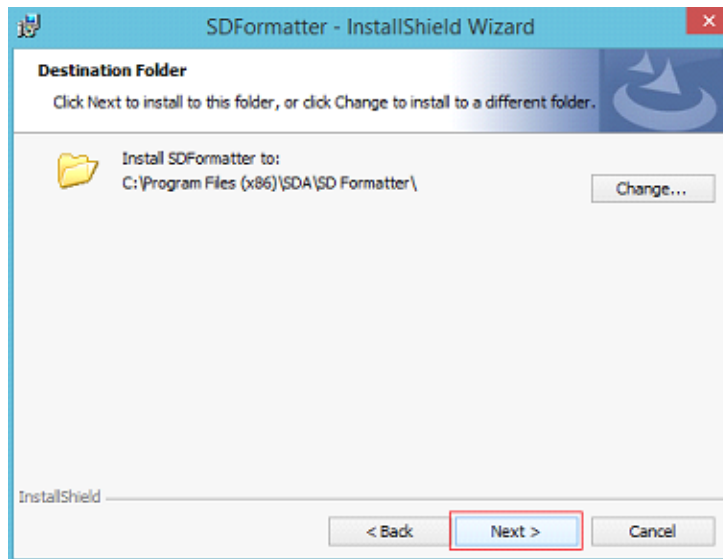
Για την χρήση της κατασκευής πρέπει να ολοκληρώσουμε τόσο το λογισμικό κομμάτι όσο και το κατασκευαστικό κομμάτι. Σε αυτό το κομμάτι της πτυχιακής μας εργασίας θα ξεκινήσουμε με την εγκατάσταση του λογισμικού. Για την εργασία αυτή θα χρειαστούμε 2 κύρια προγράμματα και ένα ειδικό εξάρτημα:

SD Formatter (Χρησιμοποιούμε το πρόγραμμα αυτό για να διαμορφώσουμε το αποθηκευτικό μέσο που χρησιμοποιεί το raspberry pi για την αναγνώριση του λειτουργικού για το raspberry pi.) Το λογισμικό για raspberry pi που λέγεται NOOBS Ένα card reader όπου διαβάζει τέτοια ειδικά μέσα αποθήκευσης (μερικοί υπολογιστές μπορεί να το έχουν πάνω στον πύργο). Αυτά τα δύο προγράμματα μπορούν να βρεθούν μέσα στον φάκελο /programms που βρίσκεται μέσα στο cd ή μπορούμε να κατεβάσουμε τις πιο πρόσφατες εκδόσεις από τις σελίδες [https://www.sdcard.org/downloads/formatter\\_4/index.html](https://www.sdcard.org/downloads/formatter_4/index.html) (για το SD Formatter) και την <https://www.raspberrypi.org/downloads/noobs/> (για το λογισμικό NOOBS). Όταν κατεβάσουμε τα παραπάνω εγκαθιστούμε το SD Format. Η διαδικασία εγκατάστασης περιγράφεται παρακάτω.



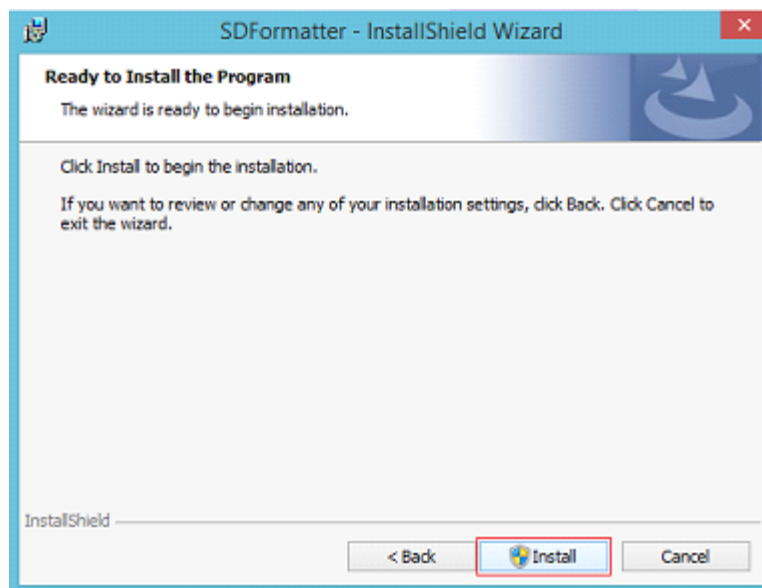
Εικόνα 50 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού(1)

Από την παραπάνω εικόνα πατάμε Next για να συνεχίσουμε την εγκατάσταση



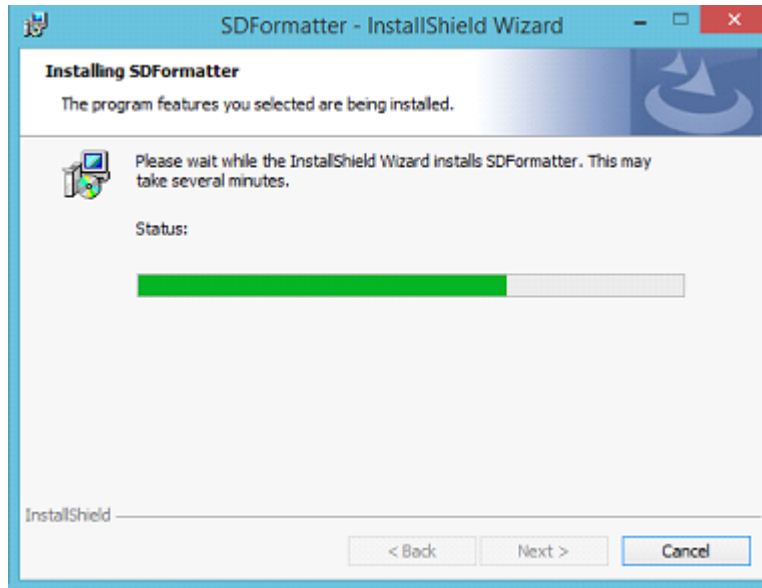
**Εικόνα 51 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού(2)**

Από την παραπάνω εικόνα μπορούμε να αλλάξουμε την διαδρομή αποθήκευσης του προγράμματος πατώντας το κουμπί Change ειδάλλως πατάμε το κουμπί NEXT για να συνεχίσουμε την εγκατάσταση.



**Εικόνα 52 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (3)**

Από την παραπάνω εικόνα μπορούμε να ξεκινήσουμε την εγκατάσταση πατώντας το πλήκτρο Install

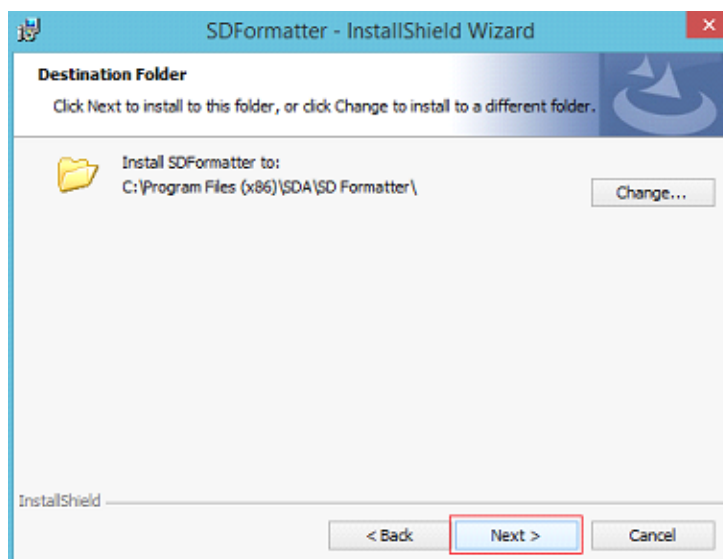


Εικόνα 53 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (4)

Όταν τελειώσουμε με την εγκατάσταση του προγράμματος στην επιφάνεια εργασίας μας πρέπει να βλέπουμε ένα εικονίδιο που γράφει SDFormatter. Είμαστε έτοιμοι για την εγκατάσταση του λογισμικού στο raspberry pi. Όπως προαναφέραμε το raspberry pi έχει μια κάρτα SD όπου εκεί αποθηκεύεται το λογισμικό λειτουργίας και όλα τα προγράμματα που εκτελεί.

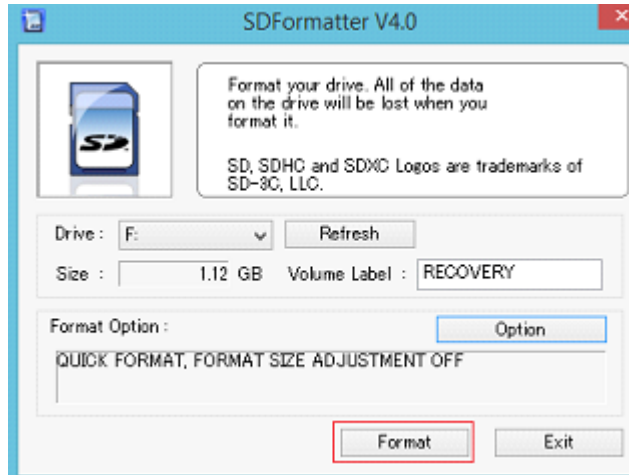
Η διαδικασία φόρτωσης του λειτουργικού προγράμματος σε αυτό το αποθηκευτικό μέσο φαίνεται παρακάτω

- Ανοίγουμε το πρόγραμμα SDFormatter



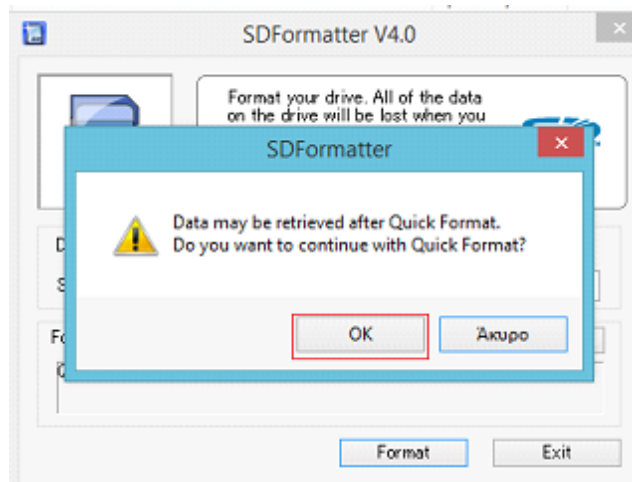
Εικόνα 54 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (5)

Από την παραπάνω εικόνα βλέπουμε την το αρχικό παράθυρο του προγράμματος SDFormatter. Από αυτό το παράθυρο επιλέγουμε Format



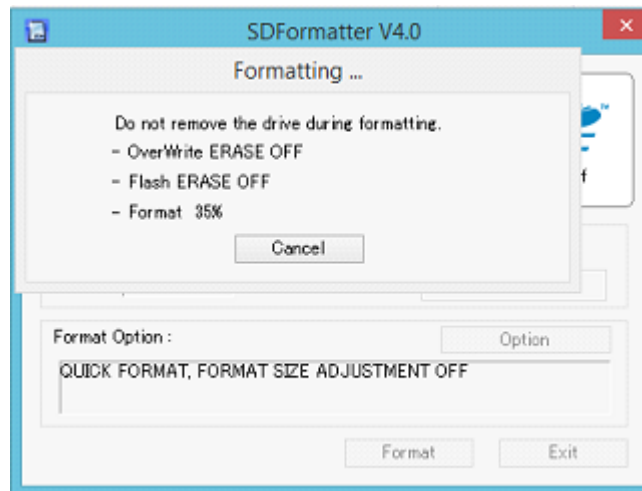
Εικόνα 55 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (6)

Από την παραπάνω εικόνα πατάμε το κουμπί OK για να συνεχίσουμε



Εικόνα 56 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (7)

Από την παραπάνω εικόνα πατάμε το κουμπί OK για να συνεχίσουμε



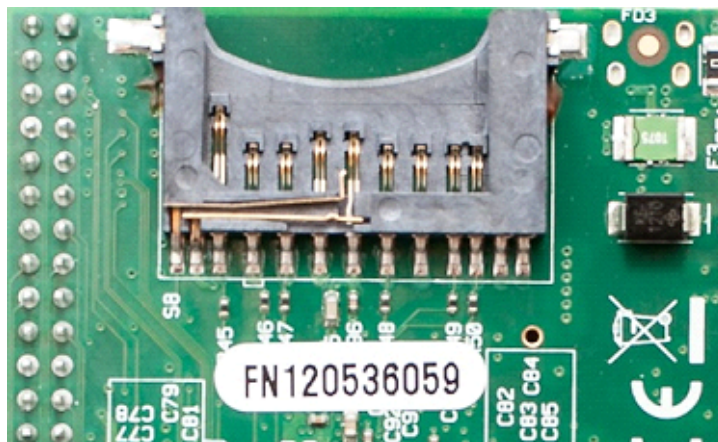
Εικόνα 57 - Προετοιμασία αποθηκευτικού μέσου για τη εγκατάσταση του λογισμικού (8)

Από την παραπάνω εικόνα βλέπουμε την διαδικασία που κάνει το σύστημά μας για την προετοιμασία του αποθηκευτικού μέσου για την εισαγωγή των αρχείων του λογισμικού raspberry pi.

Όταν τελειώσουμε με την λειτουργία του παραπάνω προγράμματος κλείνουμε το πρόγραμμα και αντιγράφουμε τα αρχεία από τον φάκελο /programms/ NOOBS\_v2\_0\_0 και τα βάζουμε(paste) στο αποθηκευτικό μέσο SD card. Είμαστε έτοιμοι να τοποθετήσουμε την SD card στο raspberry pi και να το χρησιμοποιήσουμε ως ένα σύστημα όπως ένα laptop.

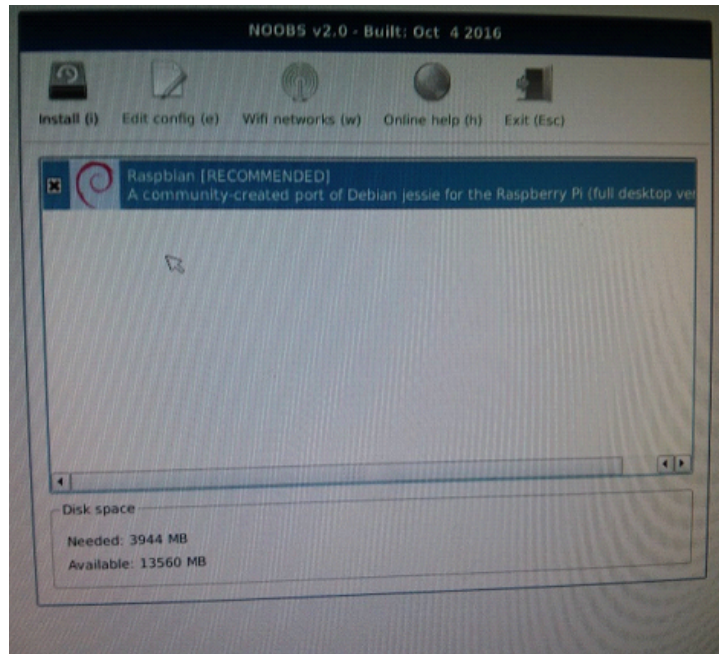
#### 4.1.2 Σύνδεση και πρώτη χρήση του raspberry pi

Αφού τοποθετήσαμε το λογισμικό του raspberry pi στον αποθηκευτικό χώρο να κάνουμε χρήση της ηλεκτρονικής πλακέτας ώστε να εγκαταστήσουμε τα απαραίτητα λογισμικά. Για να το κάνουμε αυτό τοποθετούμε την SD Card στην υποδοχή της ηλεκτρονικής πλακέτας



Εικόνα 58 - Θύρα υποδοχής κάρτας micro sd

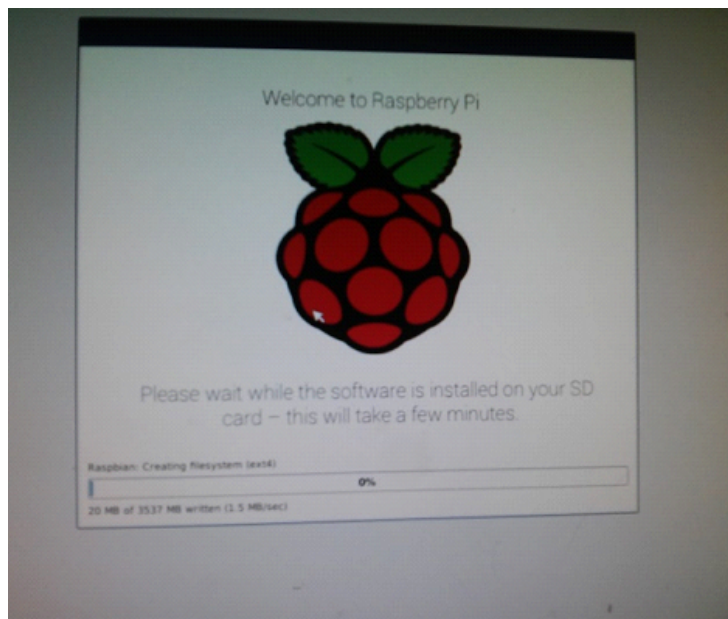
Αφού συνδέσουμε το raspberry με το ρεύμα και με κάποια οθόνη με HDMI μας εμφανίζεται το εξής παράθυρο.



**Εικόνα 59 - Εγκατάσταση λογισμικού (1)**

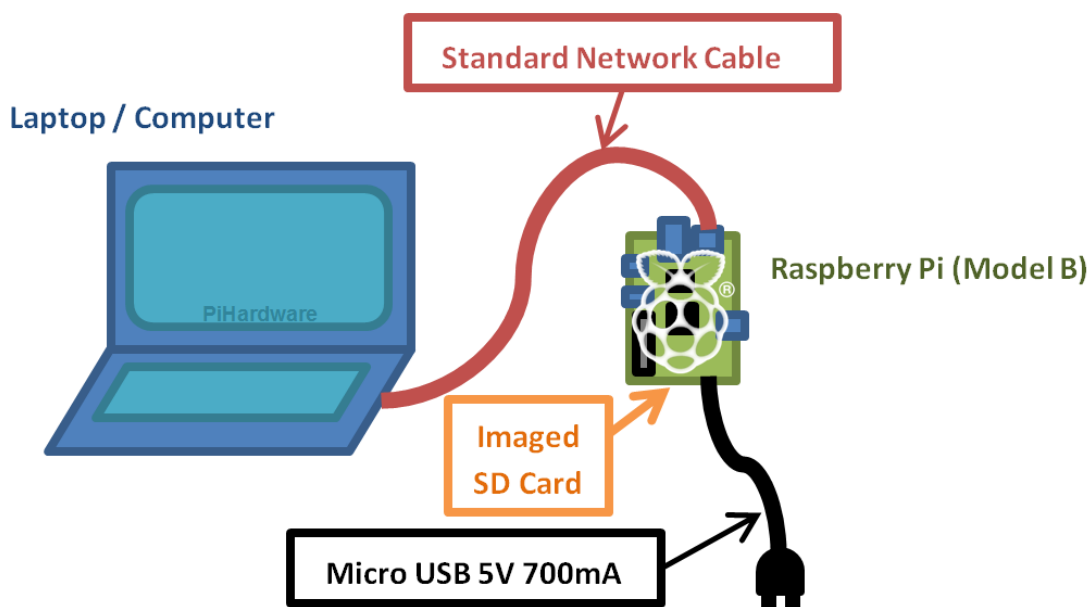
Από την παραπάνω εικόνα αφού έχουμε επιλέξει το πρώτο στοιχείο πατάμε το κουμπί Install.

Στην συνέχεια βλέπουμε την παρακάτω εικόνα όπου μας ενημερώνει ότι γίνονται οι απαραίτητες ενέργειες για την εγκατάσταση του λογισμικού και την πρώτη χρήση της πλατφόρμας



**Εικόνα 60 - Εγκατάσταση λογισμικού (2)**

Σε αυτήν την εικόνα περιμένουμε μέχρι να κάνουμε τις απαραίτητες ενέργειες. Όταν τελειώσει το πρόγραμμα την εγκατάσταση, βλέπουμε την οθόνη όπου μπορούμε να αποθηκεύσουμε αρχεία, να τα προσπελάσουμε και να πλοηγηθούμε στις δυνατότητες που μας προσφέρει το λογισμικό. Θα πρέπει να αναφέρουμε ότι υπάρχουν δύο τρόποι σύνδεσης του Raspberry pi . Ο πρώτος τρόπος είναι ο κλασικός μέσω HDMI καλωδίου κατευθείαν σε οθόνη που έχει τέτοια υποδοχή είτε μέσω κάποιου μετατροπέα από HDMI σε vga για παλιότερες οθόνες. Ο δεύτερος τρόπος είναι μέσω ssh και τον χρησιμοποιήσαμε γιατί δεν είχαμε κάποια δεύτερη οθόνη. Το **SSH** (Secure Shell) είναι ένα ασφαλές δικτυακό πρωτόκολλο το οποίο επιτρέπει τη μεταφορά δεδομένων μεταξύ δύο υπολογιστών. Το SSH όχι μόνο κρυπτογραφεί τα δεδομένα που ανταλλάσσονται κατά τη συνεδρία, αλλά προσφέρει ένα ασφαλές σύστημα αναγνώρισης καθώς και άλλα χαρακτηριστικά όπως ασφαλή μεταφορά αρχείων (SSH File Transfer Protocol, SFTP) .Το μόνο που χρειαστήκαμε είναι ένα ethernet καλώδιο για να συνδέσουμε τους δύο υπολογιστές όπως φαίνεται στην παρακάτω εικόνα και κάποια προγράμματα.

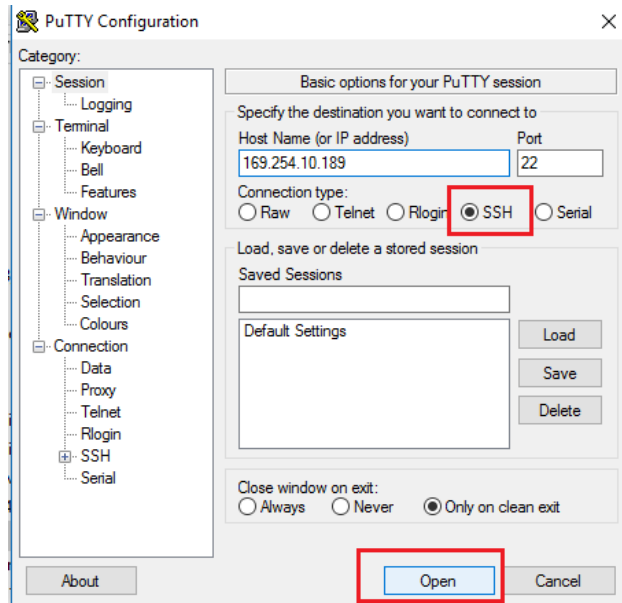


Εικόνα 61 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (1)

Όπως βλέπουμε και στην εικόνα δεν χρειάζεται επιπλέον πληκτρολόγιο και ποντίκι κάτι που κάνει την ζωή μας ευκολότερη σε συνδυασμό με την εύκολη μεταφορά αρχείων και εντολών. Το λειτουργικό του υπολογιστή μας είναι Windows 10 και χρησιμοποιήσαμε τα προγράμματα Putty και Vnc Viewer μπορεί να γίνει και σε Linux με αντίστοιχα προγράμματα.

Ανοίγουμε putty και στο host name βάζουμε την ip του Raspberry σε έμας είναι: 169.254.10.189 και πατάω open το Putty κάνει την σύνδεση μέσω του ssh όπως φαίνεται παρακάτω:

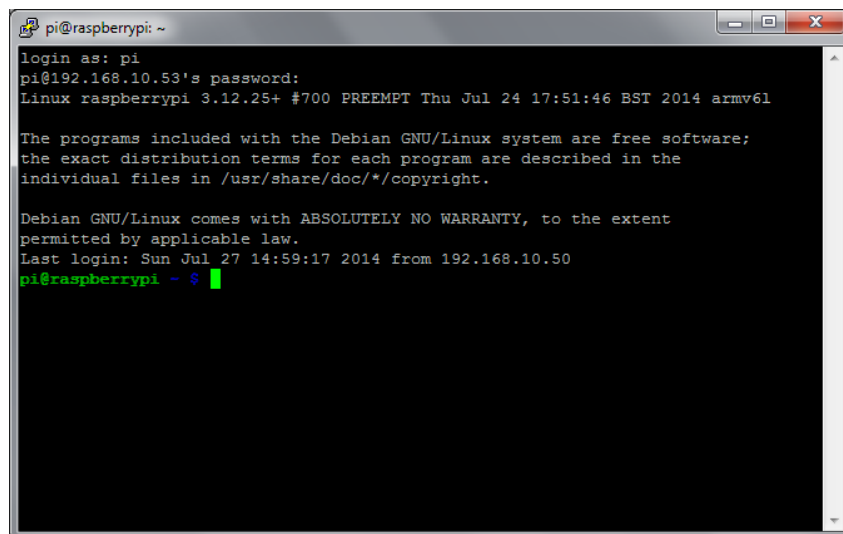




Εικόνα 62 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (2)

Μετά θα βρεθούμε στην οθόνη του τερματικού του raspberry pi και θα δώσουμε τις εντολές :

**login as:** pi  
**password:** raspberry

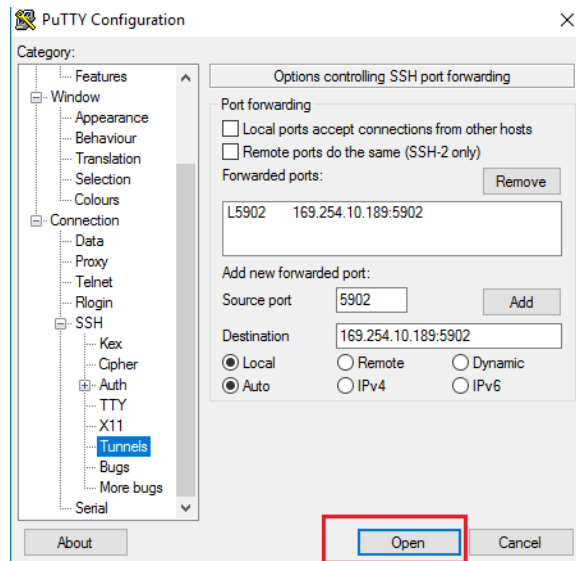


Εικόνα 63 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (3)

Είμαστε στο Raspberry ,το VNC το χρησιμοποιούμε για το γραφικό περιβάλλον. Θα δώσουμε λοιπόν στο τερματικό την εντολή για να σκοτώσουμε τυχόν γραφικά και μετά να τα ξαναδημιουργήσουμε.

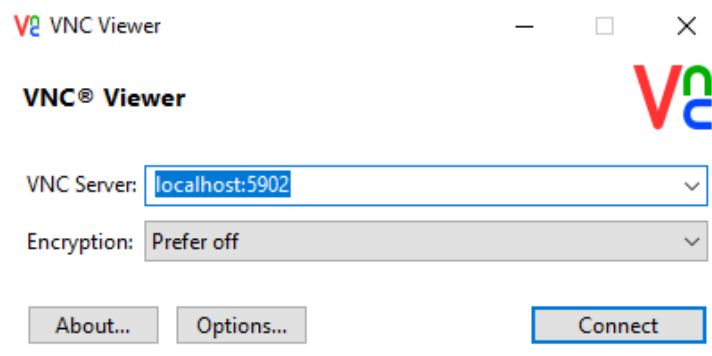
vnserver -kill : 2 (ουσιαστικά την δεύτερη οθόνη)  
 vnserver :2 (και την ανοίγουμε ξανα)

Το επόμενο βήμα είναι να ανοίξουμε δευτερο παραθυρο putty και στο host name βαζουμε την ip του raspberry.σε μενα είναι: 169.254.10.189 και στις επιλογες αριστερα παμε στο SSH και tunnels όπου βάζουμε στο destination: 169.254.10.189:5902 και στο source port: 5902 και πατάμε add και μετά open.



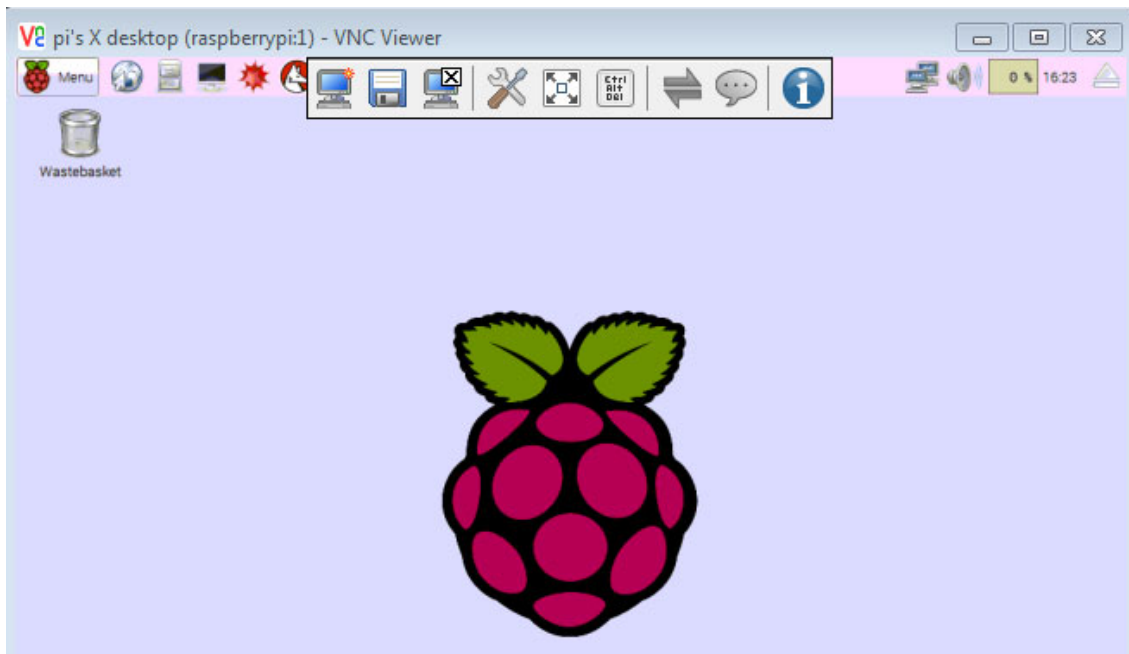
**Εικόνα 64 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (4)**

Μετά από αυτό θα βρεθούμε πάλι στην οθόνη του τερματικού και θα δώσουμε πάλι τις εντολές για να κάνουμε login . Ανοίγουμε το πρόγραμμα VNC-Viewer και στο VNC Server βαζουμε localhost:5902 και στο encryption Prefer off και παταμε connect και continue όπως φαίνεται στην παρακάτω εικόνα.



**Εικόνα 65 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (5)**

Τέλος σου ζητάει password που είναι πάλι raspberry και είσαι στο desktop του raspberry.



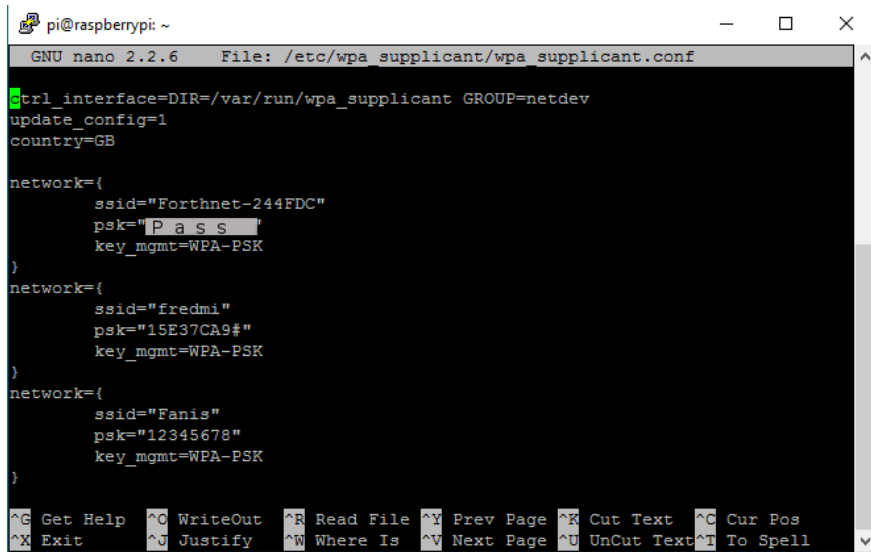
Εικόνα 66 - Διαδικασία απομακρυσμένης σύνδεσης Raspberry Pi (6)

#### 4.1.3 Σύνδεση στο Raspberry Pi μέσω του Wi-fi

Το πρώτο βήμα είναι να συνδεθούμε στο Raspberry όπως κάναμε παραπάνω. Όταν είμαστε στο Raspberry θα δώσουμε την παρακάτω εντολή.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Στην παρακάτω εικόνα βλέπουμε ότι έχουμε προσθέσει τρία δίκτυα στα οποία μπορεί να συνδεθεί το Raspberry, όταν αποθηκεύσουμε το αρχείο η σύνδεση θα πραγματοποιηθεί αυτόματα



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/wpa_supplicant/wpa_supplicant.conf

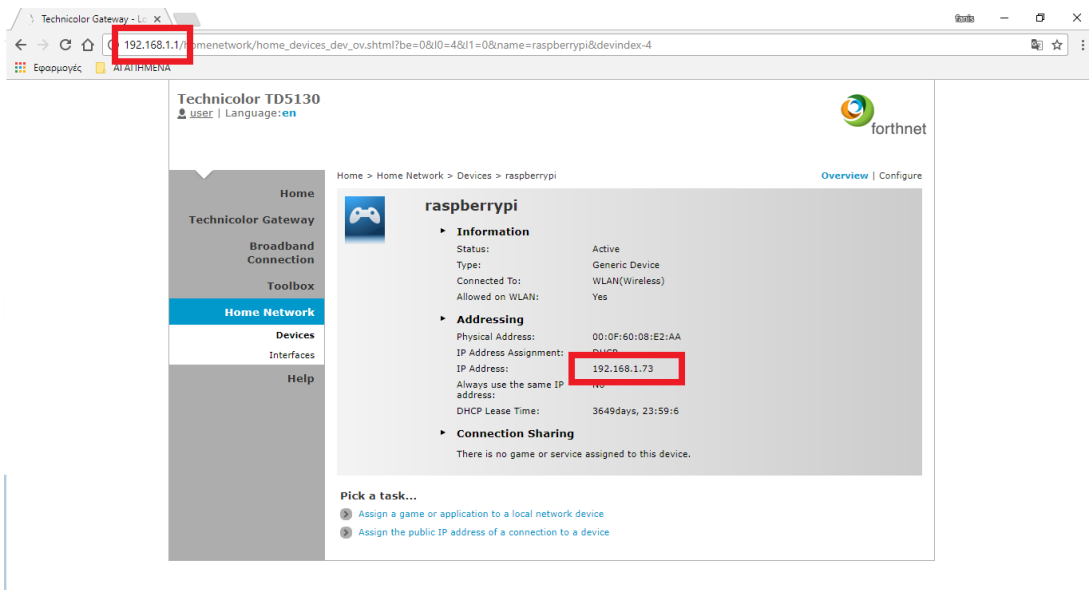
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
    ssid="Forthnet-244FDC"
    psk="P a S S"
    key_mgmt=WPA-PSK
}
network={
    ssid="fredmi"
    psk="15E37CA9#"
    key_mgmt=WPA-PSK
}
network={
    ssid="Fanis"
    psk="12345678"
    key_mgmt=WPA-PSK
}

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

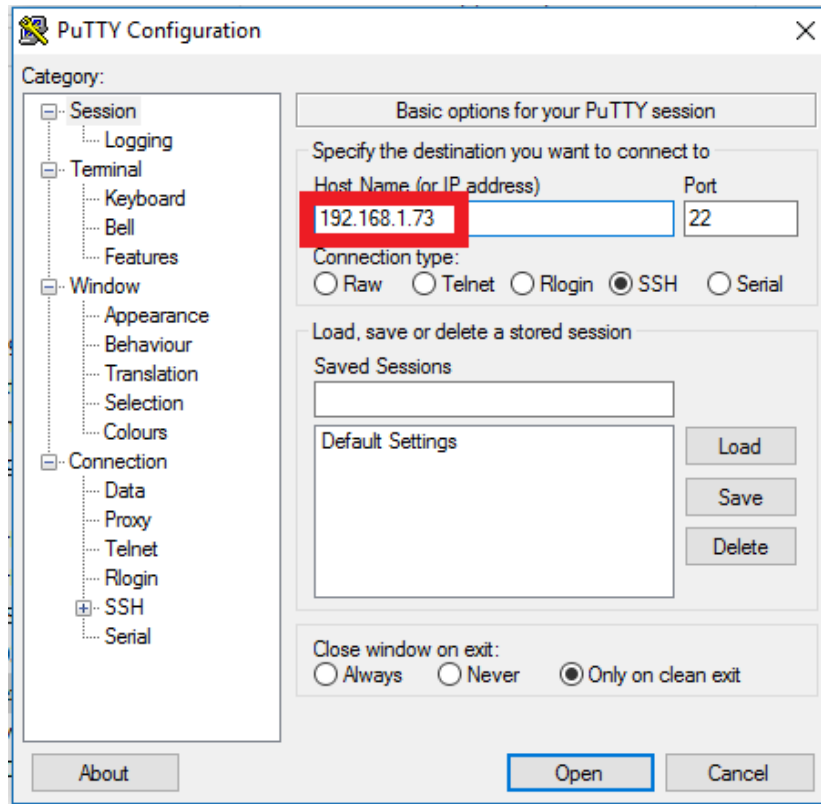
Εικόνα 67 - Διαδικασία σύνδεσης μέσω Wi-fi (1)

Επόμενο βήμα είναι να μπούμε στην σελίδα του δρομολογητή (router) μας και να βρούμε την IP του Raspberry.



Εικόνα 68 - Διαδικασία σύνδεσης μέσω Wi-fi (2)

Στο τελευταίο βήμα θα συνδεθούμε μέσω του Putty όπως κάναμε αντίστοιχα με το Ethernet.



Εικόνα 69 - Διαδικασία σύνδεσης μέσω Wi-fi (3)

#### 4.1.4 Εγκατάσταση OpenCV και Python

Αρχικά θα ανοίξουμε το τερματικό μας και θα δώσουμε τις παρακάτω εντολές για να κάνουμε Update στο Firmware.

```
sudo apt-get update
sudo apt-get upgrade
sudo rpi-update
```

Δίνουμε τις εντολές για να εγκαταστήσουμε τα κατάλληλα εργαλεία και πακέτα (πιθανόν να είναι προεγκατεστημένα)

```
sudo apt-get install build-essential cmake pkg-config
```

Εγκαθιστούμε την GTK βιβλιοθήκη για να έχουμε το GUI που χρειαζόμαστε.

```
sudo apt-get install libgtk2.0-dev
```

Εγκαθιστούμε τα πακέτα εισόδου/εξόδου βίντεο.

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

Επόμενο βήμα να εγκαταστήσουμε κάποιες βιβλιοθήκες για την βελτιστοποίηση κάποιων λειτουργιών.

```
sudo apt-get install libatlas-base-dev gfortran
```

Εγκαθιστούμε το εργαλείο pip που θα μας βοηθήσει στην εγκατάσταση κάποιων πακέτων.

```
wget https://bootstrap.pypa.io/get-pip.py  
sudo python get-pip.py
```

Με τις επόμενες εντολές θα δημιουργήσουμε το εικονικό περιβάλλον για να δουλεύουμε.

```
sudo pip install virtualenv virtualenvwrapper  
sudo rm -rf ~/.cache/pip  
  
# virtualenv and virtualenvwrapper  
export WORKON_HOME=$HOME/.virtualenvs  
source /usr/local/bin/virtualenvwrapper.sh  
  
source ~/.profile  
  
mkvirtualenv cv
```

Τώρα θα εγκαταστήσουμε τα εργαλεία της **Python 2.7**

```
sudo apt-get install python2.7-dev
```

Πρέπει επίσης να εγκαταστήσουμε το NumPy αφού οι συνδέσεις του OpenCV Python αντιπροσωπεύουν τις εικόνες ως πολυδιάστατες συστοιχίες NumPy

```
pip install numpy
```

Κατεβάζουμε και κάνουμε αποσυμπίεση το OpenCV

```
$ wget -O opencv-2.4.10.zip  
http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download  
unzip opencv-2.4.10.zip  
cd opencv-2.4.10
```

Και συνεχίζουμε με τις εξής εντολές.

```
mkdir build  
cd build  
cmake -D CMAKE_BUILD_TYPE=RELEASE -D  
CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_NEW_PYTHON_SUPPORT=ON  
-D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D  
BUILD_EXAMPLES=ON ..
```

Κάνουμε compile το OpenCV κάτι που διαρκεί αρκετή ώρα.

```
make
```

Τέλος θα εγκαταστήσουμε το OpenCV.

```
sudo make install  
sudo ldconfig
```

Κάνουμε μία σύνδεση μεταξύ του Opencv και του γραφικού περιβάλλοντός .

```
cd ~/.virtualenvs/cv/lib/python2.7/site-packages/  
ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so  
ln -s /usr/local/lib/python2.7/site-packages/cv.py cv.py
```

Και τσεκάρουμε ότι όλα δουλεύουν καλά.

```
$ workon cv  
$ python  
>>> import cv2  
>>> cv2.__version__  
'2.4.10'
```

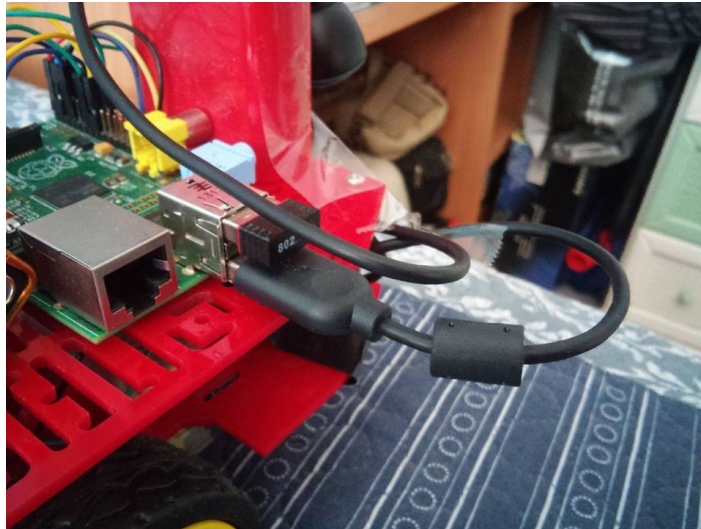
#### 4.1.5 Εγκατάσταση και σύνδεση κάμερας

Τώρα μπορούμε να εγκαταστήσουμε τα απαραίτητα προγράμματα για την επίτευξη της πτυχιακής μας.

- Εγκατάστασης της Usb camera

Για την εγκατάστασης της Usb Camera ακολουθούμε τα παρακάτω βήματα:

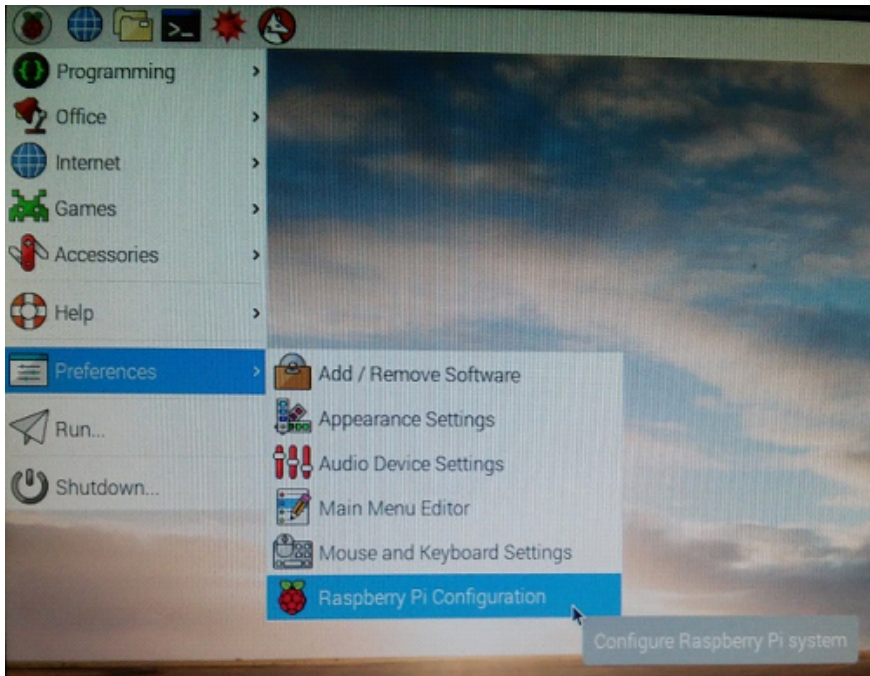
- Τοποθέτηση της Usb Camera στο Raspberry pi όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 70 - Σύνδεση Logitech Webcam c270

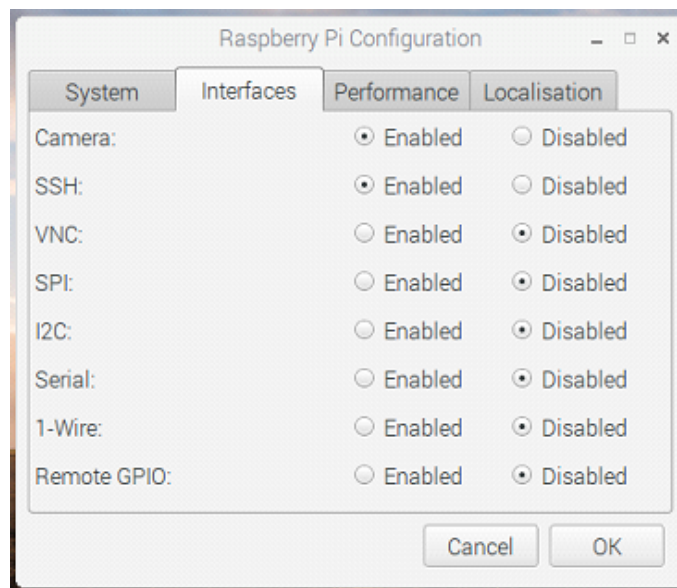
- Στην συνέχεια συνδέουμε το καλώδιο του Ethernet από την υποδοχή του raspberry pi για Ethernet, στην εξοδο διαμοιρασμού δικτύου
- Επόμενο βήμα είναι να πάμε και να ανοίξουμε το παράθυρο configuration του raspberry pi. Η παρακάτω εικόνα δείχνει πως να ανοίξουμε αυτό το παράθυρο





Εικόνα 71 - Ενεργοποίηση της κάμερας μέσω Raspberry Pi Configuration (1)

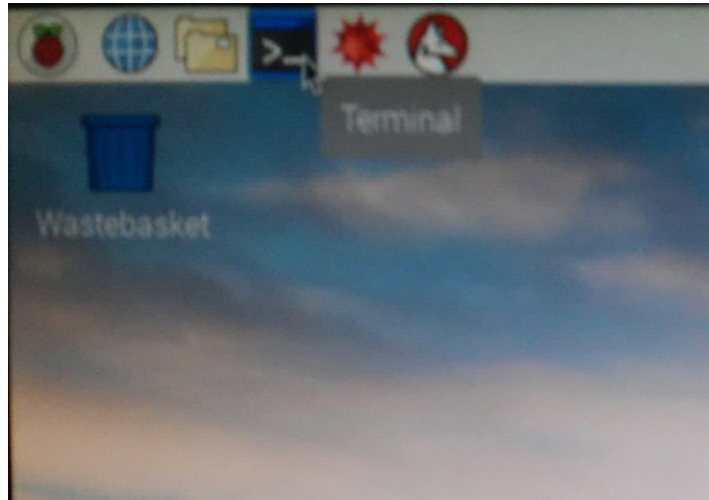
Στο παράθυρο που μας ανοίγει πατάμε το tab Interfaces και εκεί επιλέγουμε Enabled που είναι στο τμήμα Camera. Αν είναι enabled το αφήνουμε



Εικόνα 72 - Ενεργοποίηση της κάμερας μέσω Raspberry Pi Configuration (2)

Παραπάνω επιλέγουμε Enabled που είναι στο τμήμα Camera και πατάμε OK, μόλις πατήσουμε OK μας εμφανίζει ένα παράθυρο λέγοντας 'Θέλετε να γίνει επανεκίνηση τώρα', σε αυτήν την περίπτωση πατάμε Yes.

- Όταν ξεκινήσει το σύστημα ανοίγουμε το Terminal από την επιφάνεια εργασίας.



Εικόνα 73 - Terminal

- Μόλις ανοίξει το Terminal πληκτρολογούμε μια σειρά εντολών

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install python-picamera (Αυτήν την εντολή την πληκτρολογούμε  
όταν έχουμε εγκαταστήσει την έκδοση 2.x.x της γλώσσας python. )
```

Είμαστε έτοιμοι για χρήση της κάμερας.

#### 4.1.6 Εγκατάσταση drivers MPU6050 και σύνδεση στο Raspberry pi

Δεν τελειώσαμε όμως με την εγκατάσταση μας. Σε αυτήν την κατασκευή χρησιμοποιούμε ένα αισθητήρα ο οποίος λέγεται γυροσκόπιο. Ένα γυροσκόπιο μετράει την γωνιακή ταχύτητα στους τρεις άξονες x, y και z του εκάστοτε συστήματος.

Για την χρήση αυτού του αισθητήρα πρέπει να γίνει η εγκατάσταση των αντίστοιχων οδηγών(driver). Για την πραγματοποίηση αυτού του σκοπού ακολουθούμε μια σειρά από βήματα:

Ανοίγουμε ένα terminal και πληκτρολογούμε την παρακάτω εντολή Με την εντολή αυτή ανοίγουμε το αρχείο κείμενο modules για επεξεργασία ως superuser

(δηλαδή ως διαχειριστής) με την οποία παίρνουμε και τα δικαιώματα που έχει ο superuser. Αυτό σημαίνει ότι εκτελούμε το αρχείο σαν να ήμασταν διαχειριστές

```
sudo nano /etc/modules.
```

Αφού ανοίξουμε το αρχείο, στο τέλος του καταχωρούμε τις δυο παρακάτω γραμμές (αν δεν υπάρχουν)

- i2c-bcm2708
- i2c-dev

Αφού επεξεργαστούμε το αρχείο, τότε πληκτρολογούμε την ακόλουθη εντολή και σιγουρευόμαστε πως υπάρχουν οι παρακάτω δυο γραμμές

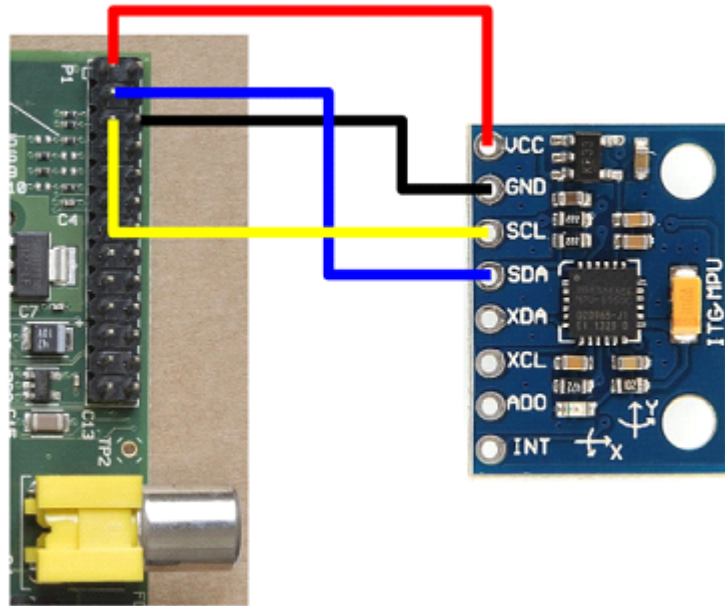
```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

- #blacklist spi-bcm2708
- #blacklist i2c-bcm2708

Κάναμε τις απαραίτητες ρυθμίσεις για την λειτουργία του αισθητήρα με το raspberry pi, τώρα είμαστε έτοιμοι για την συνδεσμολογία του αισθητήρα. Ο κάθε ακροδέκτης του αισθητήρα πάει ως εξής

Ακροδέκτης του raspberry pi	Ακροδέκτης του αισθητήρα
3.3V – Pin 1	Vcc
SDA – Pin 3	SDA
SCL – Pin 5	SCL
Ground – Pin 6	GND

Η αντιστοίχιση των ακροδεκτών μεταξύ του αισθητήρα και του raspberry pi που φαίνεται στον παραπάνω πίνακα φαίνεται και σχηματικά



Εικόνα 74 - Σχεδιάγραμμα σύνδεσης Rapsberry pi με MPU6050

Αφού έχουμε κάνει όλες τις συνδέσεις, μπορούμε να δούμε αν η πλατφόρμα μας ανιχνεύει τον αισθητήρα. Αυτό ο έλεγχος γίνεται αν πληκτρολογήσουμε την ακόλουθη εντολή

- `sudo i2cdetect -y (0 ή 1)`

Το 0 ή 1 εξαρτάται από την έκδοση της πλατφόρμας που διατίθεται (αν έχετε έκδοση 2 και επάνω χρησιμοποιείται το 1, αλλιώς το 0). Όταν εκτελέσουμε αυτήν την εντολή πρέπει να δούμε τις I2C συσκευές που ανιχνεύονται από την πλατφόρμα και σε ποια διεύθυνση επικοινωνεί με την συσκευή.

```

      0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- 68 -- -- --
70:  -- -- -- -- -- -- -- -- --
    
```

(Όπως βλέπουμε από την παραπάνω εικόνα ο αισθητήρας που έχουμε συνδέσει επικοινωνεί στην διεύθυνση 68 (δεκαεξαδικό)).

Έγιναν όλες οι απαραίτητες ενέργειες για την εγκατάσταση και την ανίχνευση του αισθητήρα από την πλατφόρμα, οπότε μπορούμε να διαβάσουμε τιμές από τον αισθητήρα και να τον συμπεριλαμβάνουμε στην ολοκλήρωση της κατασκευής.

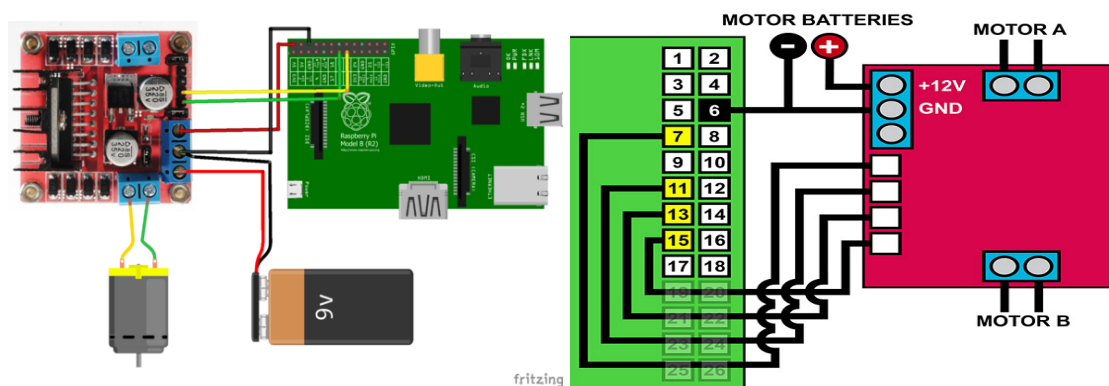
Η κατασκευή μας χωρίζεται σε τρία μέρη:

- Την ανίχνευση μιας μαύρης γραμμής
- Την κίνηση ρομπωτικού συστήματος
- Την ισορροπία του ρομπωτικού συστήματος
- Την αναγνώριση και ανάγνωση των σημάτων

Στο τέλος θα γίνει ο συνδυασμός και των τριών μερών για την επίτευξη του στόχου της εκάστοτε πτυχιακής εργασίας.

#### 4.1.7 Σύνδεση του ολοκληρωμένου L298N στο Raspberry pi

Παρακάτω βλέπουμε σε δύο σχηματικά τις συνδέσεις μεταξύ του raspberry pi και του ολοκληρωμένου.



Εικόνα 75 - Σύνδεση του ολοκληρωμένου L298N στο Raspberry Pi

Παρακάτω θα δούμε κάποια συναρτήσεις που χρησιμοποιήσαμε για τις κινήσεις .  
Κίνηση Ευθεία :

```
def forward():
    pwma.start(50)
    pwmd.start(52.8)
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(13, GPIO.HIGH)
```

```
GPIO.output(15, GPIO.LOW)
GPIO.output(12, GPIO.HIGH)
```

#### Κίνηση Πίσω :

```
def reverse():
    GPIO.output(16, GPIO.LOW)
    GPIO.output(18, GPIO.HIGH)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(15, GPIO.HIGH)
    GPIO.output(13, GPIO.LOW)
    GPIO.output(12, GPIO.HIGH)
    pwma.start(50)
    pwmd.start(52.8)
```

#### Στροφή Δεξιά :

```
def turnright():
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    pwma.start(100)
    pwmd.start(0)
    GPIO.output(13, GPIO.HIGH)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(12, GPIO.HIGH)
```

#### Στροφή Αριστερά :

```
def turnleft():
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    pwma.start(0)
    pwmd.start(100)
    GPIO.output(13, GPIO.HIGH)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(12, GPIO.HIGH)
```

#### 4.1.8 Στήσιμο Πίστας

Για την δημιουργία πίστας δοκιμών χρησιμοποιήθηκε ένα λευκό πάτωμα με μια μαύρη ταινία στη μέση. Στις παρακάτω εικόνες βλέπουμε την διαδικασία από το αρχικό στάδιο της πίστας μέχρι την τελική της μορφή.



Εικόνα 76 - Δημιουργία ζιγκ-ζαγκ με δεξιά και αριστερή γωνία 90° μοιρών



Εικόνα 77 - Πρώτη δοκιμή του οχήματος στην πίστα

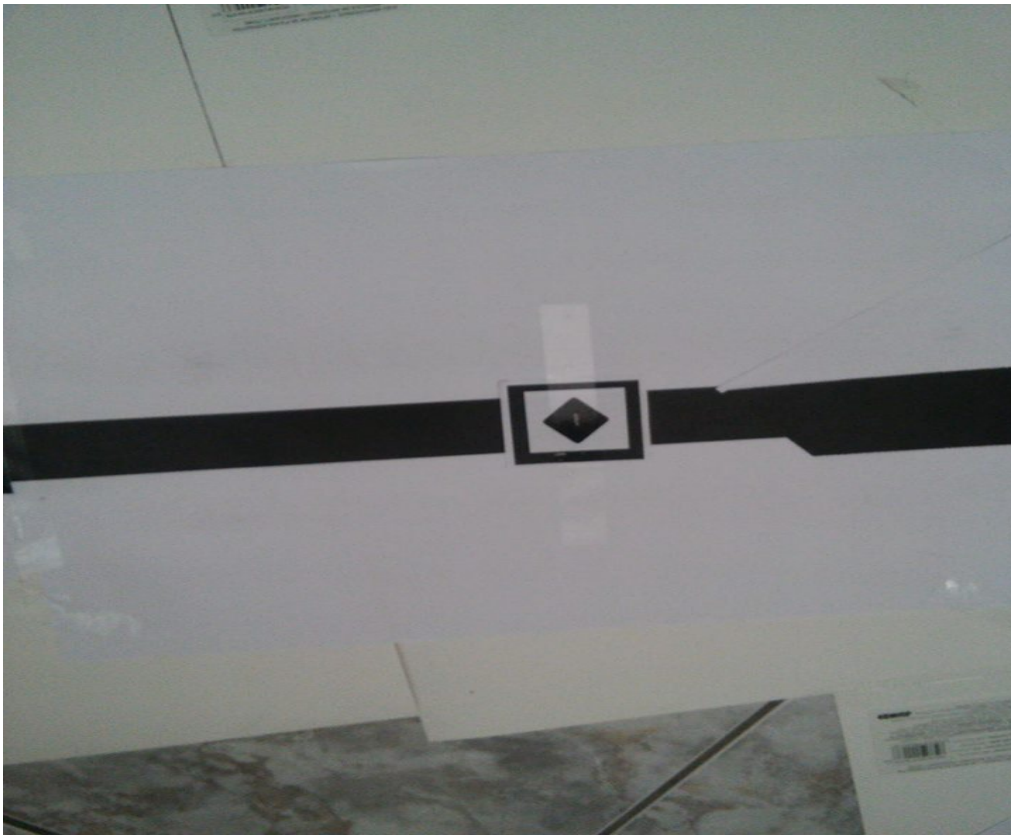


Εικόνα 78 - Αλλαγή πίστας και πρώτη βόλτα



Εικόνα 79 - Προσθήκη σημάτων





Εικόνα 80 - Το σήμα "Wait"

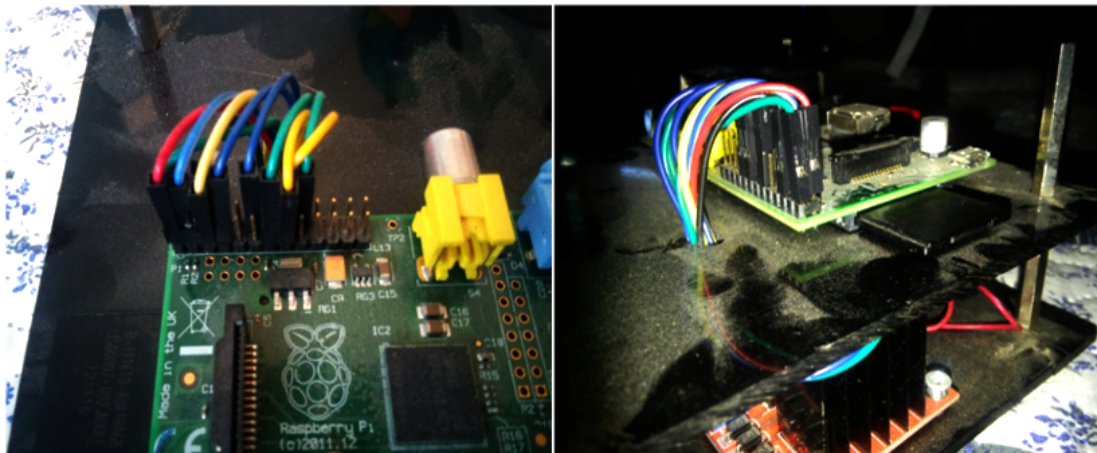


Εικόνα 81 - Το τελικό σημείο της διαδρομής το σήμα "Stop"

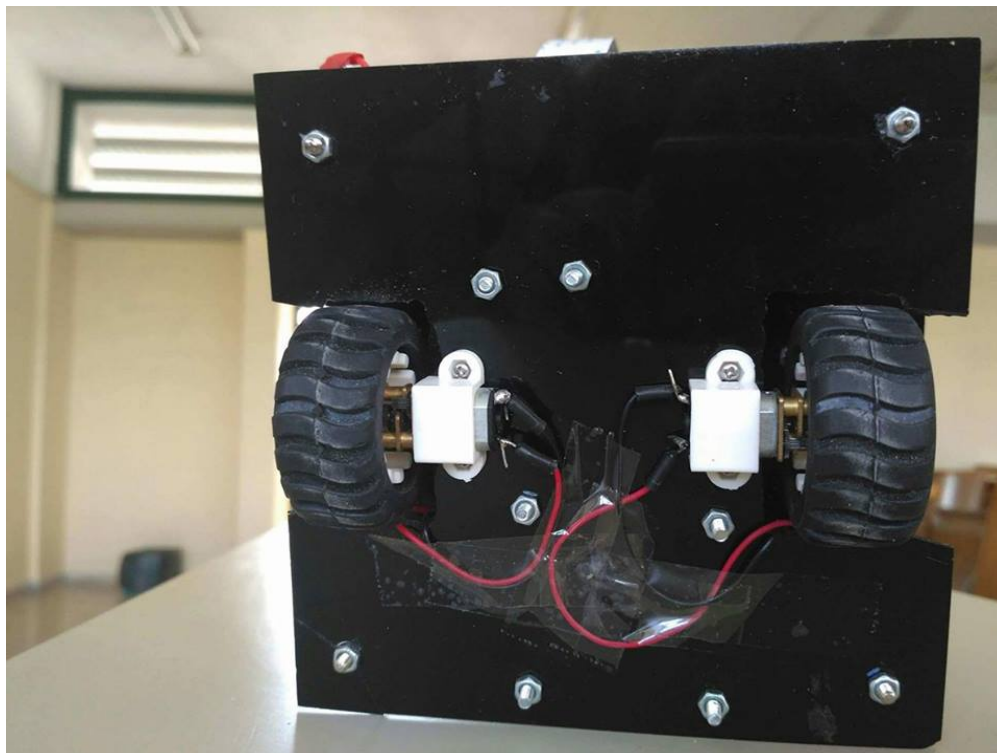
Μ' αυτόν τον τρόπο η πίστα μας πήρε την τελική της μορφή και πραγματοποιήθηκε το τελικό στάδιο των δοκιμών μας.

#### 4.1.9 Συναρμολόγηση Οχήματος

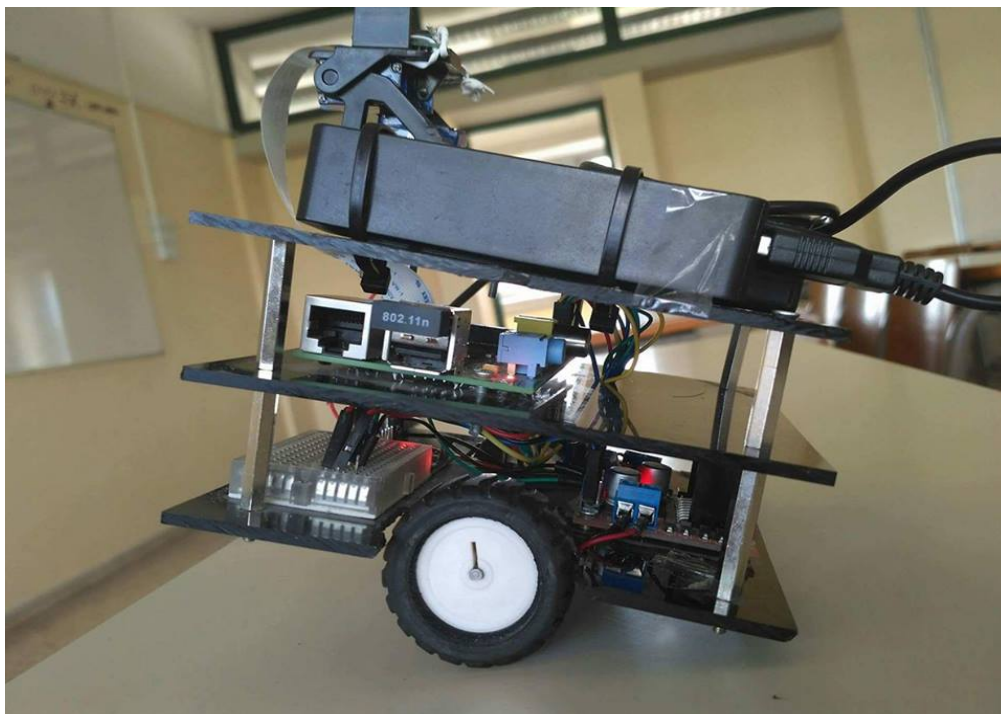
Η συναρμολόγηση του οχήματος πέρασε από διάφορα στάδια μέχρι την τελική του υλοποίηση και αυτό συνέβη γιατί αλλάξαμε το σασί και άλλων μικροπροβλημάτων που αντιμετωπίσαμε. Παρακάτω θα δούμε πρώτη φάση το πρώτο σασί και μετά το δευτεύρο δηλαδή την τελική του μορφή.



Εικόνα 82 - Συνδέσεις jumpers στις GPIO εξόδους του Raspberry Pi

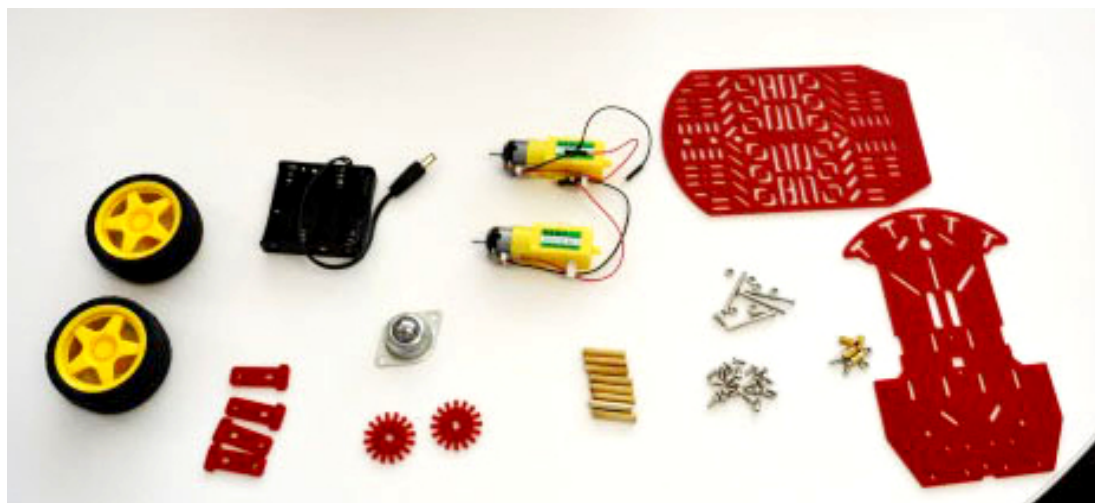


Εικόνα 83 - Συνδέσεις μοτέρ

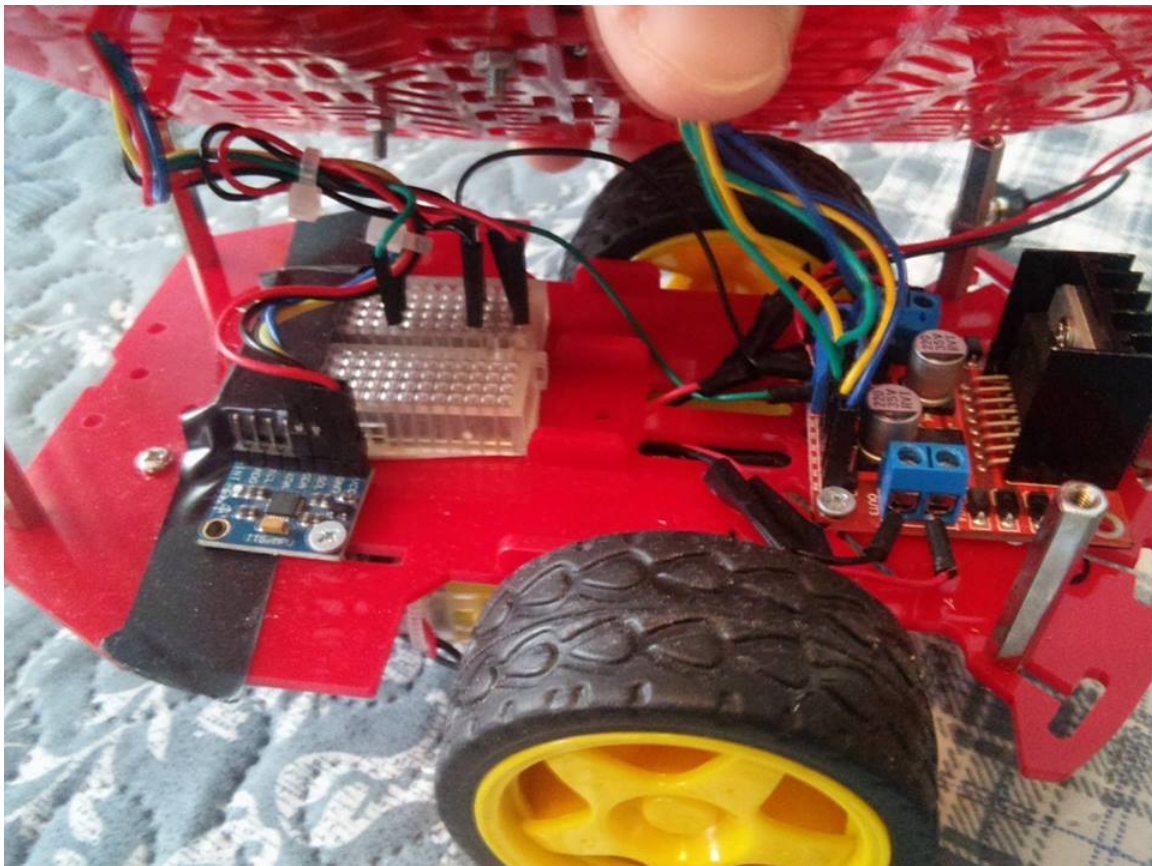


Εικόνα 84 - Τελική μορφή του πρώτου σασί

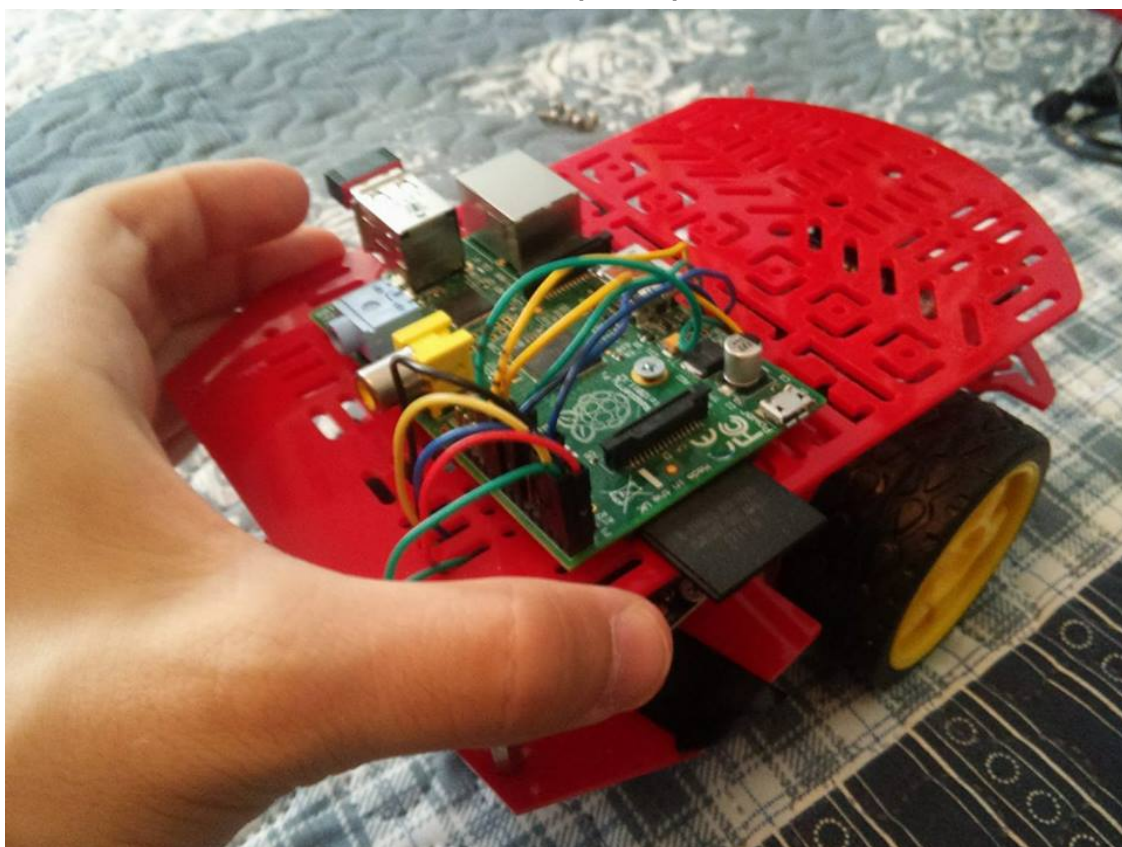
Πραγματοποιήσαμε πολλές δοκιμές μ' αυτό το σασί άλλα λόγω των δυσκολιών (κυρίως της άνισης κατανομής του βάρους). Οπότε χρησιμοποιήσαμε ένα σασί το οποίο είχαμε. Το αποσυναρμολογήσαμε αλλάξαμε τους ορθοστάτες για να έχουμε μεγαλύτερο ύψος στο πρώτο επίπεδο και συναρμολογήσαμε ξανά το όχημα μας.



Εικόνα 85 - Τα κομμάτια του σασί



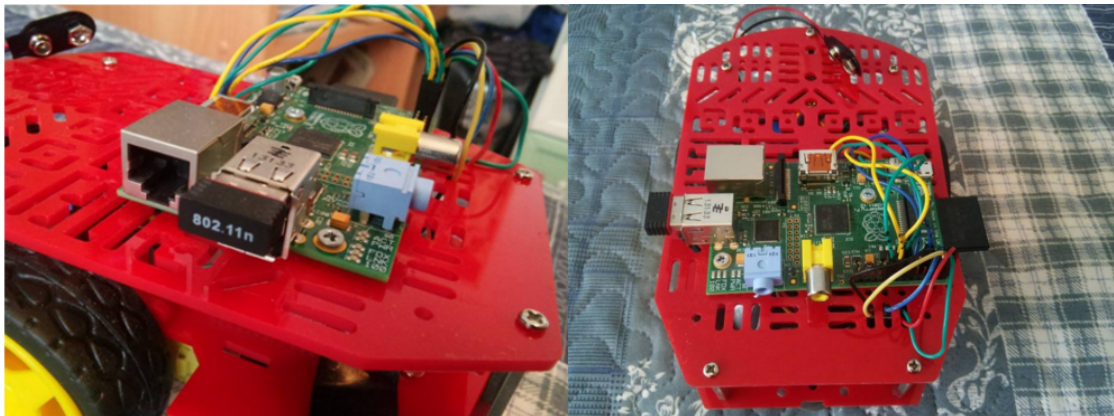
Εικόνα 86 - Το κάτω επίπεδο με το Μρι6050 και το L298η



Εικόνα 87 - Οι έξοδοι Gpio του Raspberry Pi



Εικόνα 88 - Ολοκληρωμένα τα δύο επίπεδα



Εικόνα 89 - Πριν την τοποθέτηση της κάμερας



Εικόνα 90 - Η τελική μορφή του δεύτερου σασί



## ΚΕΦΑΛΑΙΟ 5

### 5.1 Αποτελέσματα, Δυσκολίες που αντιμετωπίσαμε και Βελτιώσεις

#### 5.1.1 Αποτελέσματα Εργασίας

Τα αποτελέσματα στην πίστα της Εικόνας 79 ήταν ικανοποιητικά , το ρομπότ ακολουθούσε την σωστή πορεία σύμφωνα με την γραμμή και αντιδρούσε σωστά σύμφωνα με τα σήματα. Σε συνθήκες με χαμηλό φωτισμό μερικές φορές δεν ανταποκρινόταν στα σήματα.

#### 5.1.2 Δυσκολίες

Κατά την αποπεράτωση της πτυχιακής μας εργασίας συναντήσαμε τις παρακάτω δυσκολίες:

- Για να ρίξουμε το κόστος αγοράσαμε κάποια εξαρτήματα και υλικά από το εξωτερικό και κυρίως από την Κινέζικη αγορά με αποτέλεσμα η αναμονή να είναι μεγάλη.
- Τα καταστήματα με είδη ηλεκτρονικής ρομποτικής είναι περιορισμένα .
- Η σύνδεση ssh με καλώδιο ethernet από κάποια στιγμή και μετά δεν ήταν δυνατή και αγοράσαμε wifi adapter.
- Προχωρήσαμε στην αλλαγή του camera module με usb camera .
- Αντιμετωπίσαμε προβλήματα στην συμβατότητα μεταξύ της version της python και του opencv.
- Πρόβλημα ώστε να βρούμε τον κατάλληλο αλγόριθμο και τις κατάλληλες τιμές για την ισορροπία του οχήματος.
- Αλλαγή ολοκληρωμένου L298N όταν κήκε έπειτα από δοκιμή με υψηλή τάση .



### 5.1.3 Βελτιώσεις

Μερικές από τις βελτιώσεις που μπορούμε να πραγματοποιήσουμε για μεγαλύτερη αξιοπιστία και λειτουργικότητα είναι οι παρακάτω:

- Προσθήκη στο αυτοκινούμενο ενός distance sensor δηλαδή ενός αισθητήρα απόστασης.
- Χρήση encoder στα μοτέρ κίνησης για μεγαλύτερο έλεγχο και αξιοπιστία.
- Συγγραφή αλγορίθμου και κώδικα για την αποφυγή εμποδίων.
- Χρήση καλύτερου οχήματος .
- Χρήση αποδοτικότερων κινητήρων.
- Χρήση ισχυρότερου άρα και ταχύτερου υπολογιστικού συστήματος δηλαδή νεότερη έκδοση του raspberry pi.
- Ανάπτυξη εφαρμογή android για τον απομακρυσμένο έλεγχο.
- Μείωση του χρόνου εκτέλεσης και τερματισμού της πίστας.

## ΚΕΦΑΛΑΙΟ 6

### 6.1 Παράρτημα

#### 6.1.1 Ειδικό Λεξιλόγιο και Συντομογραφίες

Open Source - Ανοιχτού κώδικα

Scripting Language - Περιγραφική γλώσσα

Linux - Λειτουργικό σύστημα βασισμένο σε Unix

SD Formatter - Λογισμικό το οποίο χρησιμοποιείται για την εγγραφή και διαγραφή δεδομένων από και σε κάρτες SD

BCM - Broadcom SOC Channel

Password - Κωδικός πρόσβασης

OpenCV - Βιβλιοθήκη για επεξεργασία εικόνας

SSH (secure shell) - Το SSH είναι ένα πρωτόκολλο που παρέχει ασφαλή απομακρυσμένη σύνδεση σε υπολογιστές πάνω από μη ασφαλές δίκτυο.

Power Bank - Φορητή μπαταρία

Breadboard - Δοκιμαστική πλακέτα

Board - πλακέτα

IC - Ολοκληρωμένο κύκλωμα

I2C (Inter-Integrated Circuit) - Δίαυλος επικοινωνίας ολοκληρωμένων

SoC - System on Chip

## 6.1.2 Κώδικας

### cam\_and\_sings.py

```

import numpy as np
import cv2
import time
import RPi.GPIO as GPIO
#import dcmotorstest2 as motor
#Difference Variable
minDiff = 10000
minSquareArea = 5000
match = -1

#Frame width & Height
w=640
h=480

#Reference Images Display name & Original Name
ReferenceImages = ["Start.bmp", "Wait.bmp", "Stop2.bmp"]
ReferenceTitles = ["Start", "Wait5sec", "Stop"]
cap=cv2.VideoCapture(-1)
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings (False)
GPIO.setup(7, GPIO.OUT)
GPIO.setup(16, GPIO.OUT)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(12, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
pwma=GPIO.PWM(07,100)
pwmd=GPIO.PWM(12,100)
#define class for References Images
class Symbol:
    def __init__(self):
        self.img = 0
        self.name = 0

symbol= [Symbol() for i in range(3)]
#define class instances (6 objects for 6 different images)

def readReflImages():
    for count in range(3):
        image = cv2.imread(ReferenceImages[count], cv2.COLOR_BGR2GRAY)
        symbol[count].img = cv2.resize(image,(w/2,h/2),interpolation = cv2.INTER_AREA)
        symbol[count].name = ReferenceTitles[count]

```

```

def order_points(pts):
    # initialize a list of coordinates that will be ordered
    # such that the first entry in the list is the top-left,
    # the second entry is the top-right, the third is the
    # bottom-right, and the fourth is the bottom-left
    rect = np.zeros((4, 2), dtype = "float32")

    # the top-left point will have the smallest sum, whereas
    # the bottom-right point will have the largest sum
    s = pts.sum(axis = 1)
    rect[0] = pts[np.argmin(s)]
    rect[2] = pts[np.argmax(s)]

    # now, compute the difference between the points, the
    # top-right point will have the smallest difference,
    # whereas the bottom-left will have the largest difference
    diff = np.diff(pts, axis = 1)
    rect[1] = pts[np.argmin(diff)]
    rect[3] = pts[np.argmax(diff)]

    # return the ordered coordinates
    return rect

def four_point_transform(image, pts):
    # obtain a consistent order of the points and unpack them
    # individually
    rect = order_points(pts)
    (tl, tr, br, bl) = rect

    maxWidth = w/2
    maxHeight = h/2

    dst = np.array([
        [0, 0],
        [maxWidth - 1, 0],
        [maxWidth - 1, maxHeight - 1],
        [0, maxHeight - 1]], dtype = "float32")

    # compute the perspective transform matrix and then apply it
    M = cv2.getPerspectiveTransform(rect, dst)
    warped = cv2.warpPerspective(image, M, (maxWidth, maxHeight))
    return warped

def auto_canny(image, sigma=0.33):
    # compute the median of the single channel pixel intensities
    v = np.median(image)

    # apply automatic Canny edge detection using the computed median

```

```

lower = int(max(0, (1.0 - sigma) * v))
upper = int(min(255, (1.0 + sigma) * v))
edged = cv2.Canny(image, lower, upper)

# return the edged image
return edged

def resize_and_threshold_warped(image):
    #Resize the corrected image to proper size & convert it to grayscale
    #warped_new = cv2.resize(image,(w/2, h/2))
    warped_new_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    #Smoothing Out Image
    blur = cv2.GaussianBlur(warped_new_gray,(5,5),0)

    #Calculate the maximum pixel and minimum pixel value & compute threshold
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(blur)
    threshold = (min_val + max_val)/2

    #Threshold the image
    ret, warped_processed = cv2.threshold(warped_new_gray, threshold, 255,
cv2.THRESH_BINARY)

    #return the thresholded image
    return warped_processed

def signs(flag,shm):
    #Font Type
    font = cv2.FONT_HERSHEY_SIMPLEX
    #Read all the reference images
    readReflImages()

    # capture frames from the camera
    while True:
        ret, frames = cap.read()
        grays = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)
        blurreds = cv2.GaussianBlur(grays,(3,3),0)

        #Detecting Edges
        edgess = auto_canny(blurreds)

        #Contour Detection & checking for squares based on the square area
        contourss, hierarchy =
cv2.findContours(edgess,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

        for cnt12 in contourss:
            approxs = cv2.approxPolyDP(cnt12,0.01*cv2.arcLength(cnt12,True),True)

```

```

if len(approx)!=4 and flag!=0:
    main(shm)
elif len(approx)==4:
    areas = cv2.contourArea(approx)

    if areas > minSquareArea:
        warped = four_point_transform(frames, approx.reshape(4, 2))
        warped_eqs = resize_and_threshold_warped(warped)

        for i in range(3):
            diffImgs = cv2.bitwise_xor(warped_eqs, symbol[i].img)
            diffs = cv2.countNonZero(diffImgs);
            if diffs < minDiff:
                matchs = i
                print diffs
                if diffs < 4500 and diffs > 3000 and flag!=0 and shm==0:
                    print "WAIT"
                    pwma.start(0)
                    pwmd.start(0)
                    GPIO.output(16, GPIO.LOW)
                    GPIO.output(18, GPIO.LOW)
                    GPIO.output(7, GPIO.LOW)
                    GPIO.output(13, GPIO.LOW)
                    GPIO.output(15, GPIO.LOW)
                    GPIO.output(12, GPIO.LOW)
                    time.sleep(5)
                    pwma.start(30)
                    pwmd.start(32.8)
                    GPIO.output(16, GPIO.HIGH)
                    GPIO.output(18, GPIO.LOW)
                    GPIO.output(7, GPIO.HIGH)
                    GPIO.output(13, GPIO.HIGH)
                    GPIO.output(15, GPIO.LOW)
                    GPIO.output(12, GPIO.HIGH)
                    time.sleep(0.5)
                    pwma.start(0)
                    pwmd.start(0)
                    time.sleep(1.5)
                    shm = 1
                    main(shm)
                elif diffs < 6200 and diffs > 4500 and flag==0:
                    print "START"
                    pwma.start(0)
                    pwmd.start(0)
                    time.sleep(3)
                    pwma.start(0)
                    pwmd.start(0)
                    time.sleep(0.5)
                    main(shm)

```

```

elif diffs > 8000 and diffs < 9999 and flag!=0:
    print "STOP"
    GPIO.output(16, GPIO.LOW)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(7, GPIO.LOW)
    GPIO.output(13, GPIO.LOW)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(12, GPIO.LOW)
    cap.release()
    cv2.destroyAllWindows()
    GPIO.cleanup()
    diffs = minDiff
    break;

def main(shm):
    while(True):
        ret, frame = cap.read()
        # Our operations on the frame come here
        crop = frame[180:320, 0:638]
        crop2=cv2.cvtColor(crop,cv2.COLOR_BGR2GRAY)
        th,crop2 = cv2.threshold(crop2,0,255, cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
        previous = cv2.GaussianBlur(crop2, (5,5),0)

        contours, hierarchy =
cv2.findContours(crop2,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
        i=0
        for cnt in contours:
            moments = cv2.moments(cnt)
            if moments['m00']!=0:
                cx = int(moments['m10']/moments['m00']) # cx = M10/M00
                cy = int(moments['m01']/moments['m00']) # cy = M01/M00
                moment_area = moments['m00'] # Contour area from moment
                contour_area = cv2.contourArea(cnt) # Contour area using in_built function
                perimeter = cv2.arcLength(cnt, True)
                hull = cv2.convexHull(cnt)
                k = cv2.isContourConvex(cnt)
                x,y,w,h = cv2.boundingRect(cnt)
                rows,cols = previous.shape[:2]
                cv2.drawContours(previous, [cnt], 0, (0,255,0), 3)
                px = previous[cy,cx]
                if px == 255 :
                    i=i+1
                    cv2.circle(previous,(cx,cy),5,(0,0,255),-1)
            if cx < 212:
                print 'LEFT LEFT LEFT LEFT'
                if cx < 100:
                    l=100
                elif cx < 200 and cx >= 100:
                    l=75

```

```

elif cx >=200:
    l=30
    pwma.start(l)
    pwmd.start(l)
    GPIO.output(13, GPIO.HIGH)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(12, GPIO.HIGH)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(16, GPIO.LOW)
    GPIO.output(18, GPIO.HIGH)
    time.sleep(.010)
elif cx >=212 and cx <=426 :
    print 'EUTHEIA EUTHEIA EUTHEIA EUTHEIA'
    pwma.start(30)
    pwmd.start(32.8)
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(13, GPIO.HIGH)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(12, GPIO.HIGH)
    time.sleep(.01)
elif cx>426:
    print 'RIGHT RIGHT RIGHT RIGHT'
    if cx > 550 :
        p = 100
    elif cx > 400 :
        p = 75
    else:
        p = 30
    pwma.start(p)
    pwmd.start(p)
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18 ,GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(12, GPIO.HIGH)
    GPIO.output(13, GPIO.LOW)
    GPIO.output(15, GPIO.HIGH)
    time.sleep(.010)

if i!=1:
    pwma.start(30)
    pwmd.start(32.8)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(12, GPIO.HIGH)
    GPIO.output(13, GPIO.HIGH)
    GPIO.output(15, GPIO.LOW)
    GPIO.output(16, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    time.sleep(0.001)

```



```
print("TELOS TELOS TELOS")
pwma.start(0)
pwmd.start(0)
time.sleep(0.1)
flag=1
signs(flag,shm)
print 'to synolo twn shmeiwn einai:%d'%i
print 'to cx einai:%d'%cx
print 'to cy einai:%d'%cy
print 'to xrwma einai:%d' %px
GPIO.output(7, GPIO.LOW)
GPIO.output(12, GPIO.LOW)
GPIO.output(13, GPIO.LOW)
GPIO.output(15, GPIO.LOW)
GPIO.output(16, GPIO.LOW)
GPIO.output(18, GPIO.LOW)
time.sleep(.001)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

flag = 0
shm = 0
signs(flag,shm)
```

## ΚΕΦΑΛΑΙΟ 7

### 7.1 Βιβλιογραφία

<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE>

<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84>

<http://sciencing.com/main-parts-robot-7403157.html>

<https://arduinobots.wordpress.com/%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE/%CE%B5%CE%AF%CE%B4%CE%B7-%CF%81%CE%BF%CE%BC%CF%80%CF%8C%CF%84/>

<http://whatis.techtarget.com/definition/end-effector>

<http://www.robotshop.com/media/files/PDF/timeline.pdf>

<http://www.sciencekids.co.nz/sciencefacts/technology/historyofrobotics.html>

<http://mashable.com/2013/10/23/robotics-breakthroughs/#P4t56rLfsaqI>

[https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)

<https://www.raspberrypi.org/blog/raspberry-pi-zero/>

<https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>

<https://www.pi-supply.com/product/raspberry-pi-camera-board-v1-3-5mp-1080p/>

<http://blog.bitify.co.uk/2013/11/reading-data-from-mpu-6050-on-raspberry.html>

[https://el.wikipedia.org/wiki/%CE%9A%CE%AC%CF%81%CF%84%CE%B1\\_%CE%BC%CE%BD%CE%AE%CE%BC%CE%B7%CF%82](https://el.wikipedia.org/wiki/%CE%9A%CE%AC%CF%81%CF%84%CE%B1_%CE%BC%CE%BD%CE%AE%CE%BC%CE%B7%CF%82)

<https://www.engineersgarage.com/electronic-components/l293d-motor-driver-ic>

[http://nemertes.lis.upatras.gr/jspui/bitstream/10889/5555/3/Nimertis\\_Chronopoulos%28ele%29.pdf](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/5555/3/Nimertis_Chronopoulos%28ele%29.pdf)

<http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>

[http://artemis.cslab.ntua.gr/el\\_thesis/artemis.ntua.ece/DT2012-0304/DT2012-0304.pdf](http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2012-0304/DT2012-0304.pdf)

<http://apothesis.teicm.gr/xmlui/bitstream/handle/123456789/941/papariga.pdf?sequence=1&isAllowed=y>

<http://okeanis.lib.puas.gr/xmlui/bitstream/handle/123456789/3491/final.pdf?sequence=1>

<http://apothesis.teicm.gr/xmlui/bitstream/handle/123456789/876/aggelidou.pdf?sequence=1&isAllowed=y>

<http://okeanis.lib.teipir.gr/xmlui/bitstream/handle/123456789/3489/AIS0097.pdf?sequence=1>

[https://arch.ict.e.uowm.gr/docs/Python\\_Programming\\_Full\\_Book\\_Dasygenis\\_Terzidou.pdf](https://arch.ict.e.uowm.gr/docs/Python_Programming_Full_Book_Dasygenis_Terzidou.pdf)

<http://ozzmaker.com/success-with-a-balancing-robot-using-a-raspberry-pi/>

