

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

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Κίνηση Οχήματος με Raspberry Pi και Line Camera**

**Πάυλος Ε. Αλεξανδράκης**

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

**ΑΘΗΝΑ  
ΦΕΒΡΟΥΑΡΙΟΣ 2017**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Κίνηση Οχήματος με Raspberry Pi και Line Camera**

**Παύλος Ε. Αλεξανδράκης  
Α.Μ. 38468**

**Εισηγητής:**

**Δρ. Ιωάννης Έλληνας , Καθηγητής**

**Εξεταστική Επιτροπή:**

**Δρ. Πάρις Μαστοροκώστας , Καθηγητής  
Δρ. Αναστασία Βελώνη , Καθηγήτρια**

**Ημερομηνία εξέτασης:**

**03 Φεβρουαρίου 2017**

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Παύλος Αλεξανδράκης, του Ευθυμίου Αλεξανδράκη, με αριθμό μητρώου 38468 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:  
«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

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

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει 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.»

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της υλοποίησης και του σχεδιασμού ενός αυτόνομου ρομπότ με Raspberry Pi και Line Camera.

Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξε με κάθε τρόπο σε όλη την διάρκεια των σπουδών μου.



## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με τον σχεδιασμό και υλοποίηση ενός αυτοκινήτου το οποίο θα έχει τη δυνατότητα να κινείται αυτόνομα πάνω σε μία χρωματιστή κόκκινη γραμμή με τη χρήση κάμερας usb και του υπολογιστή τσέπης raspberry pi 3. Σε αυτό το πεδίο έχουν αναπτυχθεί ιστορικά αρκετοί αλγόριθμοι, κάθε ένας από τους οποίους έχει ορισμένα πλεονεκτήματα και μειονεκτήματα. Ο προτεινόμενος αλγόριθμος λαμβάνει υπ' όψη του τις απαιτήσεις του Τμήματος. Στο τέλος της πτυχιακής σκοπός είναι η παράδοση ενός αυτόνομου ρομποτικού οχήματος.

## ABSTRACT

The present thesis concerns deals with the design and implementation of a car which is able to move autonomously on a colored red line using usb camera and pocket pc raspberry pi 3. In this field historically developed several algorithms, each of which has certain advantages and disadvantages. The proposed algorithm takes account of the requirements of the Department. At the end of the graduation purpose is the delivery of an autonomous robotic vehicle.

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

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Raspberry Pi(Πλακέτα), Python(Γλώσσα Προγραμματισμού), OpenCV(Βιβλιοθήκη Επεξεργασία εικόνας), Usb Camera(Κάμερα), L293D Chip(Ολοκληρωμένο), Raspbian(Λειτουργικό)





## ΠΕΡΙΕΧΟΜΕΝΑ

|  |           |
|--|-----------|
| ΕΥΧΑΡΙΣΤΙΕΣ .....  | 5         |
| ΠΕΡΙΛΗΨΗ .....   | 7         |
| <b>ΚΕΦΑΛΑΙΟ 1.....</b>                                   | <b>13</b> |
| 1.1 Ρομπότ & Ρομποτική .....                             | 13        |
| 1.1.1 Εισαγωγή.....                                      | 13        |
| 1.1.2 Ορισμός του Ρομπότ .....                           | 14        |
| 1.1.3 Οι κατηγορίες των ρομπότ .....                     | 15        |
| 1.1.4 Το μέλλον των ρομπότ .....                         | 16        |
| 1.2 Ιστορία των κινούμενων ρομπότ .....                  | 18        |
| 1.3 Οδόμετρο του Αρχιμήδη .....                          | 24        |
| <b>ΚΕΦΑΛΑΙΟ 2.....</b>                                   | <b>29</b> |
| 2.1 Ψηφιακή Επεξεργασία Εικόνας.....                     | 29        |
| 2.1.1 Εισαγωγή.....                                      | 29        |
| 2.1.2 Ιστορική Αναδρομή .....                            | 29        |
| 2.1.3 Στόχοι .....                                       | 30        |
| 2.2 Βιβλιοθήκη OpenCV .....                              | 31        |
| 2.2.1 Τα πλεονεκτήματα της OpenCV .....                  | 34        |
| 2.2.2 Εφαρμογές της OpenCV.....                          | 35        |
| 2.2.3 Βασικές Δομές & Συναρτήσεις της OpenCV .....       | 35        |
| 2.3 Επεξεργασία Εικόνας.....                             | 37        |
| 2.3.1 Επεξεργασία με Γλώσσα Προγραμματισμού Python ..... | 37        |
| 2.3.2 Φίλτρο Gaussian .....                              | 38        |
| 2.3.3 Χρωματικό Μοντέλο HSV .....                        | 40        |
| 2.3.4 Επεξεργασία Εικόνας και Εύρεση Γραμμής.....        | 42        |
| <b>ΚΕΦΑΛΑΙΟ 3.....</b>                                   | <b>47</b> |
| 3.1 Υλικά Κατασκευής.....                                | 47        |
| 3.1.1 Raspberry Pi.....                                  | 47        |
| 3.1.2 Usb Κάμερα .....                                   | 51        |
| 3.1.3 Micro SD Card .....                                | 51        |
| 3.1.4 Ηλεκτρικός Κινητήρας.....                          | 52        |
| 3.1.5 BreadBoard .....                                   | 53        |
| 3.1.6 Ολοκληρωμένο L293D.....                            | 55        |

|   |           |
|---|-----------|
| 3.2 Λογισμικό .....                               | 56        |
| 3.2.1 Λειτουργικό Σύστημα .....                   | 56        |
| 3.2.2 Στήσιμο Μηχανημάτων .....                   | 58        |
| <b>ΚΕΦΑΛΑΙΟ 4 .....</b>                           | <b>63</b> |
| 4.1 Υλοποίηση Κατασκευής .....                    | 63        |
| 4.1.1 Στήσιμο πρώτου κυκλώματος .....             | 63        |
| 4.1.2 Στήσιμο Δεύτερου Κυκλώματος .....           | 65        |
| 4.1.3 Στήσιμο Πίστας .....                        | 67        |
| 4.1.4 Στήσιμο Οχήματος .....                      | 68        |
| <b>ΚΕΦΑΛΑΙΟ 5 .....</b>                           | <b>71</b> |
| 5.1 Αποτελέσματα Εργασίας.....                    | 71        |
| 5.2 Δυσκολίες.....                                | 71        |
| 5.3 Βελτιώσεις.....                               | 72        |
| <b>ΠΑΡΑΡΤΗΜΑ .....</b>                            | <b>73</b> |
| Ειδικό Λεξιλόγιο(Glossary) και Συνομογραφίες..... | 73        |
| Εγκατάσταση Λειτουργικού Noobs.....               | 74        |
| Εγκατάσταση Raspberry Pi.....                     | 82        |
| Εγκατάσταση OpenCV.....                           | 88        |
| Power Bank.....                                   | 90        |
| Κώδικας .....                                     | 91        |
| led.py .....                                      | 91        |
| dcmotor.py .....                                  | 91        |
| linedetect.py .....                               | 94        |
| <b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>                         | <b>99</b> |

## Κατάλογος Εικόνων

|  |    |
|--|----|
| Εικόνα 1 - Ρομπότ.....   | 14 |
| Εικόνα 2 - Οδόμετρο .....  | 24 |
| Εικόνα 3 - Μηχανισμός Λειτουργίας Οδόμετρου .....                | 26 |
| Εικόνα 4 - Ναυτικό Οδόμετρο.....                                 | 27 |
| Εικόνα 5 - Ψηφιακή Επεξεργασία Εικόνας.....                      | 29 |
| Εικόνα 6 - Εφαρμογές OpenCV .....                                | 32 |
| Εικόνα 7 - Αρχιτεκτονική OpenCV.....                             | 33 |
| Εικόνα 8 - Βασικά Συστατικά της OpenCV .....                     | 34 |
| Εικόνα 9 - Gaussian κατανομή 1-D .....                           | 38 |
| Εικόνα 10 - Gaussian κατανομή 2-D .....                          | 39 |
| Εικόνα 11 - α)Κανονική εικόνα β)Εικόνα με φίλτράρισμα .....      | 39 |
| Εικόνα 12 - Χρωματικός χώρος HSV .....                           | 40 |
| Εικόνα 13 - Μετατροπή εικόνας σε χρωματικό χώρο HSV.....         | 41 |
| Εικόνα 14 - Raspberry Pi 3 .....                                 | 49 |
| Εικόνα 15 - Usb Κάμερα .....                                     | 51 |
| Εικόνα 16 - Micro SD card .....                                  | 51 |
| Εικόνα 17 - DC Motor.....  | 53 |
| Εικόνα 18 - Breadboard .....                                     | 54 |
| Εικόνα 19 - L293D chip.....                                      | 55 |
| Εικόνα 20 - Pidora.....  | 56 |
| Εικόνα 21 - Raspbmc.....   | 57 |
| Εικόνα 22 - Raspbian.....  | 57 |
| Εικόνα 23 - Noobs.....   | 58 |
| Εικόνα 24 - Επιλογές Ρύθμισης Raspberry Pi.....                  | 60 |
| Εικόνα 25 - Κύκλωμα Raspberry Pi για ενεργοποίηση LED.....       | 63 |
| Εικόνα 26 - Κύκλωμα ενεργοποίησης LED .....                      | 64 |
| Εικόνα 27 - Κύκλωμα για τον έλεγχο δύο ηλεκτρικών κινητήρων..... | 65 |
| Εικόνα 28 - Κύκλωμα ελέγχου ηλεκτρικών κινητήρων .....           | 65 |
| Εικόνα 29 - Αριστερή/Δεξιά στροφή .....                          | 67 |
| Εικόνα 30 - Δεξιά/Αριστερή στροφή .....                          | 67 |
| Εικόνα 31 - Όχημα με Raspberry Pi & Usb Camera.....              | 68 |
| Εικόνα 32 - Υλικά Εγκατάστασης Λειτουργικού.....                 | 74 |
| Εικόνα 33 - Raspberry Pi .....                                   | 75 |
| Εικόνα 34 - Micro SD .....                                       | 75 |
| Εικόνα 35 - Απαραίτητη τάση για Raspberry pi 3.....              | 76 |
| Εικόνα 36 - SD Card Reader.....                                  | 77 |
| Εικόνα 37 - Κατέβασμα SD Formatter.....                          | 78 |
| Εικόνα 38 - Ρυθμίσεις SD Formatter .....                         | 79 |
| Εικόνα 39 - Ολοκλήρωση SD Formatter .....                        | 79 |
| Εικόνα 40 - Noobs Official Page .....                            | 80 |
| Εικόνα 41 - Αντιγραφή αρχείων στην SD card .....                 | 81 |
| Εικόνα 42 - Εξαγωγή SD card.....                                 | 81 |

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

|   |    |
|---|----|
| Εικόνα 43 - Είσοδος SD card στο Raspberry Pi..... | 82 |
| Εικόνα 44 - Θήκη Raspberry Pi.....                | 83 |
| Εικόνα 45 - Είσοδος SD card στη θήκη.....         | 83 |
| Εικόνα 46 - HDMI καλώδιο.....                     | 84 |
| Εικόνα 47 - Πληκτρολόγιο και Ποντίκι .....        | 85 |
| Εικόνα 48 - Εκκίνηση Raspberry Pi.....            | 85 |
| Εικόνα 49 - Λειτουργικά Εγκατάστασης .....        | 86 |
| Εικόνα 50 - Εκκίνηση Εγκατάστασης .....           | 87 |
| Εικόνα 51 - Επιτυχείς Εγκατάσταση.....            | 87 |
| Εικόνα 52 - Power Bank.....                       | 90 |

## ΚΕΦΑΛΑΙΟ 1

### 1.1 Ρομπότ & Ρομποτική

#### 1.1.1 Εισαγωγή

Η ρομποτική είναι μια «διεπιστημονική περιοχή» που συνδυάζει τη φυσική, την ηλεκτρολογία, τη μηχανολογία, την πληροφορική, τις τηλεπικοινωνίες, τη θεωρία συστημάτων, τον αυτόματο έλεγχο, την τεχνητή νοημοσύνη, την τεχνολογία των αισθητήριων διατάξεων (δύναμης, αφής, όρασης, κλπ), την εικονική πραγματικότητα, την επεξεργασία σήματος, την υπολογιστική όραση και την τεχνητή ζωή. Η ρομποτική ανήκει στον ευέλικτο αυτοματισμό και είναι μια τεχνολογία με μέλλον και για το μέλλον. Η ιστορική αρχή της ανιχνεύεται στο «αυτόματο ρολόι νερού» του Έλληνα μηχανικού Κτησίβιου (~300 π.Χ.) και το μηχανισμό «αυτομάτου ανοίγματος – κλεισίματος θυρών» του Έρωνα της Αλεξάνδρειας (~50 μ.Χ.). Ο όρος ρομπότ (ο οποίος προέρχεται από τη σλαβική λέξη *robota*), οφείλεται στον Τσέχο δραματικό συγγραφέα Karel Capek (1921) και σημαίνει «αναγκαστική εργασία».

Τα σημερινά ρομπότ αναπτύχθηκαν ταυτόχρονα με τον υπολογιστικό έλεγχο των εργαλειομηχανών. Το πρώτο βιομηχανικό ρομπότ κατασκευάστηκε στις ΗΠΑ από τους Devol και Engelberger και τέθηκε σε λειτουργία το 1960 σε ένα χυτήριο μετάλλων. Από τότε, στα πενήντα χρόνια που ακολούθησαν, η ανάπτυξη της ρομποτικής ήταν ραγδαία και σήμερα βρίσκονται σε δράση παγκοσμίως πάνω από 120.000 ρομπότ (με πρωταγωνιστές τις ΗΠΑ, την Ιαπωνία και την Ευρώπη) που καλύπτουν βιομηχανικές, ιατρικές, επιστημονικές και κοινωνικές εφαρμογές.

[1]

### 1.1.2 Ορισμός του Ρομπότ

Σύμφωνα με το Robot Institute of America, ως ρομπότ μπορούμε να ορίσουμε έναν μηχανισμό σχεδιασμένο ώστε, μέσω προγραμματιζόμενων κινήσεων, να μεταφέρει υλικά, τεμάχια, εργαλεία ή ειδικευμένες συσκευές, με σκοπό την επιτέλεση ποικιλίας εργασιών. Ένας τέτοιος μηχανισμός περιλαμβάνει συνήθως τις ακόλουθες συνιστώσες :



**Εικόνα 1 - Ρομπότ**

- Ένα μηχανολογικό υποσύστημα, το οποίο ενσωματώνει τη δυνατότητα του ρομπότ για εκτέλεση έργου. Το υποσύστημα αυτό αποτελείται από μηχανισμούς που επιτρέπουν στο ρομπότ να κινείται, όπως αρθρώσεις, συστήματα μετάδοσης κίνησης, επενεργητές – κινητήρες, οδηγούς, κτλ.

- Ένα υποσύστημα αίσθησης, μέσω του οποίου το ρομπότ συγκεντρώνει πληροφορίες για την κατάσταση στην οποία βρίσκεται τόσο το ίδιο, όσο και το περιβάλλον του. Το υποσύστημα αυτό, εκτός των άλλων, είναι υπεύθυνο για την αποδοχή των εξωτερικών εντολών, την επεξεργασία τους, την μετάφρασή τους σε ηλεκτρική ισχύ που δοθεί στους κινητήρες του ρομπότ, καθώς επίσης και για την παραγωγή σημάτων εξόδου που θα πληροφορούν για την κατάσταση του συστήματος. Στο υποσύστημα αίσθησης περιλαμβάνονται όργανα μέτρησης, αισθητήρες, ηλεκτρονικά στοιχεία κτλ.

- Ένα σύστημα ελέγχου, το οποίο συνδυάζει κατάλληλα την αίσθηση με τη δράση, έτσι ώστε το ρομπότ να λειτουργεί αποτελεσματικά και με τον επιθυμητό τρόπο. Ο ελεγκτής του ρομπότ επιβλέπει και συντονίζει ολόκληρο το σύστημα, για τη σχεδίαση και την υλοποίησή του δε, απαιτείται ο συνδυασμός γνώσεων από πολλές γνωστικές περιοχές, όπως είναι ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών κτλ.

### 1.1.3 Οι κατηγορίες των ρομπότ

Ως ρομπότ χαρακτηρίζεται κάθε ελεγχόμενη από υπολογιστή μηχανή που μπορεί να εκτελέσει εργασίες τις οποίες κάνει ο άνθρωπος. Τα ρομπότ της πρώτης γενιάς δεν είχαν ικανότητα υπολογισμού και αίσθησης, ενώ τα ρομπότ της 2ης γενιάς διαθέτουν περιορισμένη υπολογιστική ικανότητα. Τα ρομπότ της 3ης γενιάς διαθέτουν «νοημοσύνη» (είναι όπως λέμε «έξυπνα ρομπότ») και μπορούν να λύνουν προβλήματα και να παίρνουν αποφάσεις κατά τη διάρκεια της εργασίας τους. Τις ικανότητες αυτές αποκτούν με τεχνικές «τεχνητής νοημοσύνης» και «αίσθησης».

Το ρεπερτόριο των εφαρμογών διευρύνεται συνεχώς σε νέα πεδία της ανθρώπινης δραστηριότητας και οι επιστήμονες συνεχίζουν αδιάκοπα την προσπάθεια ανάπτυξης και κατασκευής αληθινά «έξυπνων ρομπότ» τα οποία να μπορούν να συμπεριφέρονται, όσο γίνεται πιο πολύ, όπως και ο άνθρωπος.

Στη λαϊκή αντίληψη τα ρομπότ συνδέθηκαν από τη δεκαετία του 1960 με ταινίες όπως «Η τελευταία ημέρα του κόσμου» και «Ο πόλεμος των άστρων». Χαρακτηριστικά παραδείγματα ρομπότ με συγκεκριμένους ρόλους ανθρώπινης μορφής είναι τα ρομπότ R2D2 και C3PO.

Σήμερα υπάρχει ένας θαυμαστός κόσμος ρομπότ που μπορούν να μετακινούνται, να βαδίζουν, να βλέπουν, να ομιλούν και να εκτελούν λεπτούς χειρισμούς που απαιτούν εξυπνάδα και επιδεξιότητα. Ο κόσμος αυτός διαρκώς



εξαπλώνεται και ικανοποιεί τις ανάγκες εργασίας, παραγωγής, υγείας, ευημερίας και ψυχαγωγίας του ανθρώπου. Οι πέντε βασικές κατηγορίες ρομπότ είναι:

- Βιομηχανικά ρομπότ
- Κινητά ρομπότ
- Ιατρικά ρομπότ
- Τηλερομπότ
- Κοινωνικά ρομπότ

[2]

#### **1.1.4 Το μέλλον των ρομπότ**

Τα σύγχρονα ρομπότ έχουν μηχανικές και νοητικές ικανότητες που προηγουμένως ανήκαν στη σφαίρα επιστημονικής φαντασίας. Η ανάπτυξή τους θα συνεχίσει να επεκτείνεται με ολοένα νέα είδη ρομπότ κατάλληλα για τη βιομηχανία, την επιστήμη και την καθημερινή ζωή του ανθρώπου. Ήδη σήμερα υπάρχουν ιπτάμενα μη επανδρωμένα οχήματα-ρομπότ, ρομποτικοί οδηγοί (ρομποτικά μπαστούνια) τυφλών, ρομπότ ποδοσφαιριστές, πολύποδα ρομπότ ανίχνευσης ηφαιστειών, σμήνη συνεργαζόμενων ρομπότ, ρομποτικά έντομα, κοκ. Η έρευνα και ανάπτυξη προς την κατεύθυνση αυτή συνεχίζεται αδιάκοπα έχοντας ως βασική προτεραιότητα την ασφάλεια και άνεση του ανθρώπου και την απαλλαγή του από δύσκολες, επίπονες και επικίνδυνες εργασίες.

Στο μέλλον ο άνθρωπος πιθανώς θα ζει ανάμεσα στα ρομπότ στο σπίτι, το δρόμο, την εργασία, το νοσοκομείο, κλπ και θα συμβιώνει με αυτά για την επιτυχία του κοινού στόχου υγείας, υψηλής ποιότητας ζωής και μακροζωίας.

Ο άνθρωπος φανταζόταν πάντοτε μηχανές που να του μοιάζουν και να τον υπηρετούν στην καθημερινή του ζωή. Στην ελληνική μυθολογία είναι γνωστός ο Τάλως, το πρώτο ρομπότ που κατασκευάστηκε από τον Δαίδαλο (ή Ήφαιστο) με

εντολή του Δία και δόθηκε στον Μίνωα για την προστασία της Κρήτης από τους εχθρούς της.

Σήμερα σχεδιάζονται και κατασκευάζονται «έξυπνα ρομπότ» για τη βιομηχανία, τις υπηρεσίες, τις οικιακές εργασίες, αλλά δυστυχώς και για πολεμικούς σκοπούς. Όπως και άλλα επιτεύγματα της επιστήμης (λ.χ. της πυρηνικής φυσικής και τεχνολογίας) έχουν χρησιμοποιηθεί εναντίον του ανθρώπου, έτσι και τα ρομπότ μπορούν να χρησιμοποιηθούν (και έχουν χρησιμοποιηθεί) για καταστροφικούς σκοπούς. Οι αυτό καθοδηγούμενες βόμβες λέιζερ δεν είναι παρά «έξυπνα ρομπότ».

Ο αρχικός φόβος ότι ο αυτοματισμός θα αυξήσει την ανεργία αποδείχθηκε στην πράξη ότι δεν ισχύει, γιατί ο αυτοματισμός (μηχανοποίηση, ρομποτική, πληροφορική) δημιούργησαν πολύ περισσότερα νέα επαγγέλματα από όσα εξαφάνισαν. Έτσι συνολικά ο αυτοματισμός οδήγησε σε μείωση της ανεργίας, η οποία όμως εξαρτάται από την οικονομική κατάσταση τόσο στο επίπεδο κάθε χώρας όσο και διεθνώς.

Η πορεία της ρομποτικής δείχνει ότι αυτή θα συνεχίσει να αναπτύσσεται για πολύ μεγάλο χρονικό διάστημα. Στην ανάπτυξη αυτή μπορούν και πρέπει να συμβάλλουν φυσικοί, μαθηματικοί, μηχανικοί, πληροφορικοί, αλλά και γιατροί, χημικοί, βιολόγοι, και οικονομοτεχνικοί επιστήμονες.

[3]

## 1.2 Ιστορία των κινούμενων ρομπότ

Στην παράγραφο αυτήν, θα κάνουμε μία εκτεταμένη αναφορά στην ιστορία των κινούμενων ρομπότ, η οποία και έχει στιγματίσει την πορεία της τεχνολογίας και της επιστήμης, από τα μέσα του 20ου αιώνα μέχρι σήμερα. Η εξέλιξη των κινούμενων ρομπότ προέκυψε με σκοπό την επιτέλεση διάφορων λειτουργιών, από την εκτέλεση απλών καθημερινών εργασιών, μέχρι τη διενέργεια πολύπλοκων ερευνών, ενώ σήμερα χρησιμοποιούνται και στις πλέον μεγάλης σημασίας διαστημικές αποστολές.

1939 – 1945: Κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου, εμφανίστηκαν τα πρώτα κινούμενα ρομπότ, σαν αποτέλεσμα της τεχνολογικής προόδου ενός αριθμού από νέα σχετικά ερευνητικά πεδία, όπως η Επιστήμη Υπολογιστών ή η Κυβερνητική.

1948 – 1949: Ο William Grey Walter κατασκεύασε τα ρομπότ Elmer και Elsie, δυο αυτόνομα ρομπότ, τα οποία ονομάστηκαν Machina Speculatrix, επειδή τους άρεσε να εξερευνούν το περιβάλλον τους. Το κάθε ένα από τα ρομπότ Elmer και Elsie ήταν εξοπλισμένο με έναν αισθητήρα φωτός. Αν έβρισκαν κάποια πηγή φωτός κινούνταν προς αυτήν, αποφεύγοντας παράλληλα τα εμπόδια που βρίσκονταν στον δρόμο τους. Αυτά τα ρομπότ αποδείκνυαν ότι μία πολύπλοκη συμπεριφορά μπορούσε να προκύψει από έναν απλό σχεδιασμό.

1961 – 1963: Το Johns Hopkins University ανέπτυξε το ρομπότ 'Beast'. Το ρομπότ Beast για να κινηθεί χρησιμοποιούσε ένα sonar. Όταν οι μπαταρίες του αποφορτιζόνταν, τότε το ρομπότ θα εντόπιζε μία βάση φόρτισης με την οποία και θα συνδεόταν.

1969: Το Mowbot ήταν το πρώτο ρομπότ το οποίο κούρευε αυτόματα το γρασίδι.

1970: Το Cart line follower του Πανεπιστημίου Stanford, ήταν ένα κινούμενο ρομπότ το οποίο ήταν ικανό να ακολουθήσει μία άσπρη γραμμή, χρησιμοποιώντας μία κάμερα. Ήταν ασύρματα συνδεδεμένο με έναν μεγάλο σύστημα υπολογιστή, ο οποίος και πραγματοποιούσε τους υπολογισμούς.

Την ίδια περίπου περίοδο (1966–1972), το Ερευνητικό Κέντρο του Stanford κατασκευάζει και πραγματοποιεί έρευνα πάνω σε ένα ρομπότ που ονομάστηκε Shakey, εξαιτίας της σπασμωδικής του κίνησης. Το Shakey ήταν εξοπλισμένο με μία κάμερα, αισθητήρα μέτρησης της απόστασης, αισθητήρες επαφής καθώς επίσης και κεραία για ασύρματη επικοινωνία. Ήταν το πρώτο ρομπότ που επιχειρηματολογούσε για τις πράξεις του. Αυτό σημαίνει ότι δίνονταν στο ρομπότ πολύ γενικές εντολές, και εκείνο θα αντιλαμβανόταν τα απαραίτητα βήματα που έπρεπε να κάνει, ώστε να φέρει εις πέρας την ζητούμενη εργασία.

Τέλος, η Σοβιετική Ένωση εξερευνάει την επιφάνεια της Σελήνης, με το Lunokhod 1, ένα σεληνιακό ρόβερ.

1976: Με το πρόγραμμα Viking, η NASA στέλνει δύο μη επανδρωμένα διαστημικά σκάφη στον Άρη.

1980: Το ενδιαφέρον του κοινού γύρω από τα ρομπότ αυξάνεται, και αυτό έχει ως αποτέλεσμα την σταδιακή παραγωγή ρομπότ τα οποία μπορούν να αγοραστούν για οικιακή χρήση. Τα ρομπότ αυτά εξυπηρετούσαν ψυχαγωγικούς ή εκπαιδευτικούς σκοπούς. Παραδείγματα αποτελούν το ρομπότ RB5X (το οποίο υπάρχει και σήμερα) και η σειρά HERO (Heathkit Educational RObot). Επίσης, το Cart του Stanford είναι πλέον ικανό να πλοηγείται μεταξύ εμποδίων και να χαρτογραφεί το περιβάλλον του.

Αρχές 1980: Η ομάδα του Ernst Dickmanns κατασκευάζει στο Bundeswehr University του Μονάχου, τα πρώτα ρομπότ – αυτοκίνητα, τα οποία είχαν τη δυνατότητα να κινηθούν έως και 55mph σε άδειους δρόμους.

1987: Το HRL (Hughes Research Laboratories) κατασκευάζει την πρώτη, χαρτογραφημένη και βασισμένη σε αισθητήρες, αυτόνομη λειτουργία ενός ρομποτικού οχήματος.

1989: Ο Mark Tilden επινοεί την ρομποτική BEAM (Biology, Electronics, Aesthetics, Mechanics).

Αρχές 1990: Ο Joseph Engelberger, πατέρας του βιομηχανικού ρομποτικού βραχίονα, εργάζεται με συναδέλφους προκειμένου να σχεδιάσει τα πρώτα εμπορικά διαθέσιμα αυτόνομα κινούμενα ρομπότ για νοσοκομειακή χρήση (τα οποία επρόκειτο να βγουν στην αγορά από την Helpmate).

Την ίδια περίοδο, το Υπουργείο Άμυνας των ΗΠΑ επενδύει πάνω στο πρόγραμμα MDARS-I, το οποίο βασίζεται στα ρομπότ ασφάλειας εσωτερικών χώρων.

1991: Ο Edo Franzl, ο Andre Guignard και ο Francesco Mondada ανέπτυξαν το Khepera, ένα μικρό αυτόνομο κινούμενο ρομπότ, που έχει ως σκοπό ερευνητικές δραστηριότητες. Το συγκεκριμένο πρόγραμμα υποστηρίχτηκε από το εργαστήριο του Πολυτεχνείου της Λωζάννης, LAMI- EPFL lab.

1993 – 1994: Τα Dante I και Dante II αναπτύχθηκαν από το Carnegie Mellon University. Και τα δύο ήταν βαδίζοντα ρομπότ που είχαν σκοπό την εξερεύνηση ενεργών ηφαιστειών.

1994: Τα 2 δίδυμα ρομπότ – οχήματα VaMP της Daimler-Benz και VITA-2 της UniBwM, ταξίδεψαν (με επιβαίνοντες σε αυτά) περισσότερα από 1000km σε αυτοκινητόδρομο του Παρισιού 3 λωρίδων και με φυσιολογικά υψηλή κίνηση, με ταχύτητες κοντά στα 130 km/h. Τα ρομπότ αυτά επιδείκνυαν αυτόνομη οδήγηση σε ελεύθερες λωρίδες, οδήγηση σε φάλαγγα αυτοκινήτων, καθώς και αλλαγές λωρίδας (αριστερά και δεξιά), με αυτόνομη προσπέραση άλλων αυτοκινήτων.

1995: Το ημιαυτόνομο αυτοκίνητο ALVINN ήταν κατευθυνόμενο υπό τον έλεγχο ηλεκτρονικού υπολογιστή, για μία πολύ μεγάλη διαδρομή. Παρόλα αυτά, γκάζι και φρένο ήταν ελεγχόμενα από άνθρωπο οδηγό.

1995: Την ίδια χρονιά, ένα από τα ρομπότ – αυτοκίνητα του Ernst Dickmanns (με αυτοελεγχόμενα γκάζι και φρένα) ταξίδεψε περισσότερα από 1000 μίλια, από το Μόναχο στην Κοπεγχάγη, και επέστρεψε, εν μέσω κίνησης, με ταχύτητες κοντά στα 120 m/h, εκτελώντας περιστασιακά ελιγμούς για να προσπεράσει άλλα αυτοκίνητα (μόνο σε κάποιες κρίσιμες περιπτώσεις, ένας οδηγός ασφαλείας ανέλαβε τον χειρισμό). Ενεργής όραση χρησιμοποιήθηκε για να αντιμετωπίσει τις ξαφνικές αλλαγές σε ορισμένα σημεία του δρόμου.

1995: Το προγραμματιζόμενο κινούμενο ρομπότ της Pioneer γίνεται εμπορικά διαθέσιμο σε μία προσιτή τιμή, επιτρέποντας την εκτεταμένη ανάπτυξη στην έρευνα και την πανεπιστημιακή μελέτη της ρομποτικής μέσα στην επόμενη δεκαετία, καθώς η ρομποτική κίνησης γίνεται βασικό μέρος του κύκλου σπουδών πολλών πανεπιστημίων.

1996 – 1997: Η NASA στέλνει στον Άρη το Mars Pathfinder μαζί με το ρόβερ του, το Sojourner, το οποίο είχε ως σκοπό την εξερεύνηση της επιφάνειας, δεχόμενο εντολές από τη Γη. Το Sojourner ήταν εξοπλισμένο με σύστημα αποφυγής κινδύνων, το οποίο επέτρεπε στο ρόβερ να βρίσκει αυτόνομα το δρόμο του μέσα στο άγνωστο έδαφος του Άρη.

1999: Η Sony παρουσιάζει τον Aibo, έναν ρομποτικό σκύλο με δυνατότητα να βλέπει, να βαδίζει και να αλληλεπιδρά με το περιβάλλον του. Επίσης, παρουσιάζεται το τηλεκατευθυνόμενο στρατιωτικό ρομπότ PackBot.

2001: Ξεκινάει το Swarm-bots project. Τα Swarm-bots μοιάζουν με αποικίες εντόμων, και αποτελούνται από έναν μεγάλο αριθμό μικρών ανεξάρτητων ρομπότ, τα οποία αλληλεπιδρούν μεταξύ τους και όλα μαζί εκτελούν πολύπλοκες εργασίες.

2002: Εμφανίζεται το Roomba, ένα οικιακό αυτόνομο κινούμενο ρομπότ που καθαρίζει το πάτωμα.

2003: Η Axxon Robotics αγοράζει την Intellibot, κατασκευάστρια εταιρία μίας σειράς εμπορικών ρομπότ τα οποία τρίβουν και σκουπίζουν τα πατώματα νοσοκομείων, γραφείων κτλ. Τα ρομπότ περιποίησης δαπέδου από την Intellibot Robotics, λειτουργούν εντελώς αυτόνομα, χαρτογραφώντας το περιβάλλον τους και χρησιμοποιώντας ένα σύνολο αισθητήρων για την αποφυγή εμποδίων κατά την πλοήγησή τους.

2004: Γίνεται εμπορικά διαθέσιμο το Robosapien, ένα βιομετρικό ρομπότ – παιχνίδι σχεδιασμένο από τον Mark Tilden. Επίσης, με το 'The Centibots Project', 100 αυτόνομα ρομπότ λειτουργούν μαζί, προκειμένου να καταρτίσουν τον χάρτη ενός άγνωστου χώρου, και ψάχνουν για αντικείμενα μέσα στον χώρο αυτόν. Τέλος, στον πρώτο διαγωνισμό DARPA Grand Challenge, πλήρως αυτόνομα οχήματα συναγωνίζονται, σε έναν κενό χώρο.

2005: Η Boston Dynamics δημιουργεί ένα τετράποδο ρομπότ, προκειμένου να κουβαλάει βαριά φορτία, σε έδαφος που είναι πολύ ανώμαλο για οχήματα.

2006: Η Sony σταματάει την κατασκευή του Aibo, και η Helpmate διακόπτει την παραγωγή της, αλλά το χαμηλότερου κόστους, αυτόνομο ρομπότ – βοηθός PatrolBot γίνεται διαθέσιμο, στην συνεχόμενη προσπάθεια να γίνουν τα κινούμενα ρομπότ εμπορικά βιώσιμα. Το Υπουργείο Άμυνας των ΗΠΑ διακόπτει το πρόγραμμα MDARS-I, αλλά επενδύει στο MDARS- E, ένα αυτόνομο ρομπότ ειδικά κατασκευασμένο για την εκτέλεση γεωργικών εργασιών. Επίσης, παρουσιάζεται το TALON-Sword, το πρώτο εμπορικά διαθέσιμο ρομπότ με εκτοξευτή χειροβομβίδων και άλλες ενσωματωμένες οπλικές επιλογές. Τέλος, το Asimo της Honda εκπαιδεύεται να τρέχει και να ανεβαίνει σκάλες.

2007: Γράφεται ιστορία με το DARPA Urban Grand Challenge, όπου έξι οχήματα ολοκληρώνουν αυτόνομα μία πολύπλοκη πορεία, η οποία περιλάμβανε άλλα επανδρωμένα οχήματα και εμπόδια. Εξαπλώνονται τα έξυπνα ρομπότ στην διανομή εργασιών, της Kiva Systems. Αυτές οι ευφυείς μονάδες ταξινόμησης, κατατάσσουν τη λειτουργία τους ανάλογα με τη δημοτικότητα του περιεχομένου της.

Το Tug γίνεται ένα δημοφιλές μέσο για την μετακίνηση νοσοκομειακών αποθεμάτων μεγάλου φορτίου, ενώ το Speci-Minder και το Motivity ξεκινούν τη μεταφορά αίματος και άλλων δειγμάτων των ασθενών, από τους σταθμούς των νοσοκόμων σε διάφορα εργαστήρια.

Το Seekur, το πρώτο ευρέως διαθέσιμο, μη στρατιωτικό, βοηθητικό ρομπότ εξωτερικών χώρων, έχει τη δυνατότητα να ρυμουλκήσει ένα τριών τόνων όχημα από μία θέση παρκινγκ, να το οδηγήσει αυτόνομα σε έναν εσωτερικό χώρο και εν συνεχεία, να καθοδηγηθεί και πάλι προς τα έξω. Εν τω μεταξύ, το PatrolBot εκπαιδεύεται να ακολουθεί ανθρώπους και να εντοπίζει πόρτες που είναι μισάνοιχτες.

2008: Η Boston Dynamics κυκλοφορεί το απόσπασμα ενός βίντεο, για μίας νέας γενιάς BigDog ικανό να βαδίζει σε παγωμένο έδαφος και να ανακτά την ισορροπία του σπρώχνεται από το πλάι.

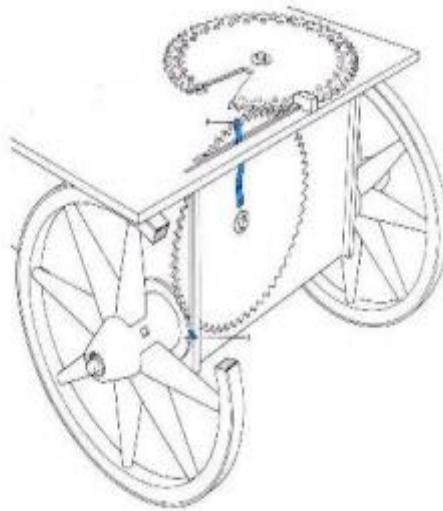
2010: Ο διαγωνισμός Multi Autonomous Ground-robotic International Challenge (MAGIC) παρουσιάζει ομάδες αυτόνομων οχημάτων που χαρτογραφούν ένα μεγάλο αστικό περιβάλλον, που αναγνωρίζουν και παρακολουθούν ανθρώπους, και που αποφεύγουν εχθρικά αντικείμενα.

[4]



### 1.3 Οδόμετρο του Αρχιμήδη

Ένα από τα πιο βασικά ζητήματα της παρούσας διπλωματικής εργασίας, είναι η συνεχόμενη εκτίμηση της θέσης ενός κινούμενου ρομπότ. Η πραγματοποίηση της συγκεκριμένης διαδικασίας θα γίνει, όπως θα δούμε στα επόμενα κεφάλαια, με τεχνικές οδομετρίας, δηλαδή με συνεχόμενη μέτρηση της απόστασης που διανύεται από το ρομποτικό όχημα. Οι τεχνικές αυτές βασίζονται στη λογική του Οδομέτρου του Αρχιμήδη, το οποίο και θα περιγράψουμε στην συνέχεια, μαζί με μία σύντομη ιστορική αναδρομή.



**Εικόνα 2 - Οδόμετρο**

Το οδόμετρο θεωρείται ο πρόδρομος του σημερινού κοντέρ, του οργάνου που μετρά τη χιλιομετρική απόσταση, ενώ εφευρέτης του ήταν ο μεγαλύτερος μαθηματικός της αρχαιότητας, ο Αρχιμήδης. Το οδόμετρο που περιγράφει ο Ρωμαίος αρχιτέκτονας και μηχανικός Βιτρούβιος, κατασκευάστηκε τον 1ο αιώνα π.χ. και είναι μηχανισμός που προσαρμοζόταν στους τροχούς μιας άμαξας και κατέγραφε την απόσταση που αυτή είχε διανύσει. Περιλαμβάνει ξύλινους οδοντωτούς τροχούς κι αυτή είναι η βασική του διαφορά από το οδόμετρο που περιγράφει ο Ήρωνας, το οποίο χρησιμοποιεί ατέρμονες κοχλίες.

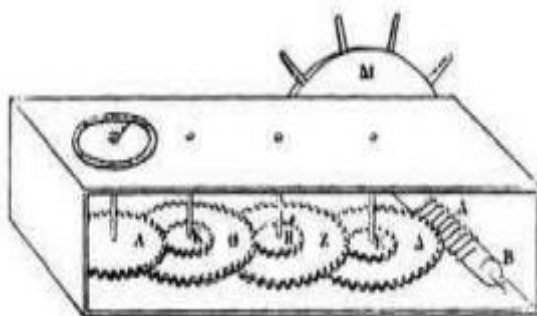
Πρώτη αναφορά, λοιπόν, στο οδόμετρο γίνεται από τον Βιτρούβιο, που αφιερώνει και ειδικό τμήμα γι' αυτό χωρίς όμως να παρέχεται καμία απεικόνιση του οργάνου. Με την ανάγνωση αυτού του τμήματος γίνεται αμέσως φανερό ότι ο συγγραφέας περιγράφει κάποιο μηχάνημα προγενέστερο της εποχής του, του οποίου πιθανότατα είχε δει μόνο τα υπολείμματα και γι' αυτό δεν το απεικόνισε. Ίσως επρόκειτο για κάποια μορφή άμαξας που διέθετε έναν δυσανάλογα μεγάλο τροχό, αναγκαίο για το οδόμετρο και κατά πάσα πιθανότητα δεν επρόκειτο για χρηστικό μηχάνημα, γνώμη με την οποία συμφωνεί και ο D.Price σε σχετική εργασία του. Αυτή η ασάφεια ως προς την περιγραφή και η ανυπαρξία κάποιου σχεδίου είναι μια από τις αιτίες που έχει καθυστερήσει η έρευνα γύρω από το μηχανισμό αυτό. Μολονότι ο Βιτρούβιος δεν ονομάζει κάποιον ως τον εφευρέτη και κατασκευαστή του μηχανισμού, εντούτοις η σύγχρονη έρευνα έχει αποδώσει την εφεύρεση του οργάνου στον Αρχιμήδη, δεδομένου ότι υπάρχει και μια αναφορά από τον ιστοριοδίφη Ιωάννη Τζέτζη σχετικά με την ανακάλυψή του από τον Συρακούσιο.

Το όργανο αυτό το γνωρίζουμε επακριβώς χάρη στην περιγραφή του Ήρωνα του Αλεξανδρινού, μεγάλου μαθηματικού και μηχανικού της ελληνιστικής περιόδου, την οποία ενσωμάτωσε στο έργο του (όταν είχε διατελέσει διευθυντής του περίφημου Μουσείου της Αλεξάνδρειας, ενός επιστημονικού κέντρου εφάμιλλου με τα σημερινά πολυτεχνεία). Το μηχανικό οδόμετρο ή αλλιώς και «δρομόμετρο», αποτελείται από ένα σύμπλεγμα οδοντωτών τροχών (γρاناζιών), που με την βοήθεια ελίκων μεταφέρουν την κίνηση του οχήματος και την μετατρέπουν σε μονάδες μέτρησης. Έτσι είναι εύκολο σε οποιονδήποτε να ενημερωθεί για την απόσταση που έχει διανύσει το όχημα, συμβουλευόμενος την διαβαθμισμένη πλάκα που υπάρχει στην επάνω πλευρά του κιβωτίου το οποίο περικλείει το μηχανισμό, ένα μηχανισμό που σύγχρονοι ερευνητές ονόμασαν «ταξίμετρο», όπως φαίνεται και στην παρακάτω εικόνα.



τους επιστήμονες που «διασταύρωσαν τα ξίφη» τους έντονα μετά την ανακάλυψη και μελέτη του περίφημου υπολογιστή των Αντικυθήρων, ανακατασκεύασε στα πλαίσια αυτής της διαμάχης το ναυτικό δρομόμετρο στηριζόμενος στην περιγραφή του Ήρωνα.

Ο συγγραφέας διατύπωσε την άποψη ότι το ναυτικό δρομόμετρο εξελίχθηκε κυρίως στην Αλεξάνδρεια, ναυτική πρωτεύουσα της εποχής, όπου έδρασε ο Ήρων, και υποστήριξε ότι τα σημερινά δρομόμετρα έχουν μικρή διαφορά από τα αρχαία. Με βάση την περιγραφή του Ήρωνα, ο Κ. Ν. Ράδος κατασκεύασε ένα όμοιο μηχανήμα από ξύλο και χαλκό, το οποίο εξέθεσε στη Διεθνή Έκθεση του Μπορντώ το 1907, μαζί με άλλα αντικείμενα σχετικά με την ιστορία του ελληνικού ναυτικού και της ελληνικής ναυτιλίας, από τους αρχαίους χρόνους. Η συλλογή εκείνη απέσπασε μεγάλο έπαινο.



**Εικόνα 4 - Ναυτικό Οδόμετρο**

Αυτά είναι όλα όσα γνωρίζουμε για το οδόμετρο («ταξίμετρο») του Ήρωνα και το αντίστοιχο ναυτικό οδόμετρο (ή δρομόμετρο). Του οφείλουμε ότι διέσωσε αρχικά τα στοιχεία που υπήρχαν για το μηχανισμό αυτό, ο οποίος πιθανότατα ανακαλύφθηκε από τον Αρχιμήδη, και σε δεύτερη βάση του οφείλουμε τη σημαντική βελτίωση του οργάνου αλλά και την προσαρμογή του για χρήση στην θάλασσα. Το ναυτικό δρομόμετρο, με τις βελτιώσεις του Ήρωνα, έγινε ένα εύχρηστο και αναγκαίο όργανο για την μέτρηση των ναυτικών αποστάσεων και συνέβαλε στη γνώση της ναυτικής τέχνης. Και αυτά από τον 1ο μ.Χ. αιώνα, μια

περίοδο κατά την οποία βρισκόταν σε εξέλιξη ένας μεγάλος αριθμός εμπορικών και εξερευνητικών αποστολών εκ μέρους των Ελλήνων σε Ανατολή, Δύση, Βορρά και Νότο. Πρόκειται για την πρώτη περίοδο των εξερευνήσεων και ανακαλύψεων, η οποία χρονολογείται ανάμεσα στα 300 π.Χ. – 300 μ.Χ. και επέκτεινε τα όρια του γνωστού τότε κόσμου.

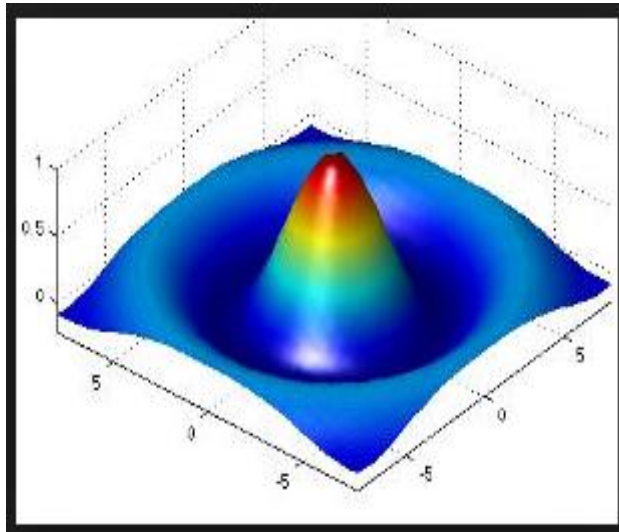
[5]

## ΚΕΦΑΛΑΙΟ 2

### 2.1 Ψηφιακή Επεξεργασία Εικόνας

#### 2.1.1 Εισαγωγή

Η ψηφιακή επεξεργασία εικόνας (Image Processing) είναι η χρήση αλγορίθμων για να παρουσιάσουν την επεξεργασία εικόνας σε ψηφιακές εικόνες. Ως υπό επίπεδο της ψηφιακής επεξεργασίας σήματος η ψηφιακή επεξεργασία εικόνας έχει πολλά πλεονεκτήματα σε σχέση με την αναλογική επεξεργασία, επιτρέπει ένα πολύ μεγαλύτερο εύρος αλγορίθμων να εφαρμοστούν σαν δεδομένα εισόδου, και μπορεί να αποτρέψει προβλήματα όπως η συγκέντρωση θορύβου και διαστρέβλωση σημάτων κατά τη διάρκεια της επεξεργασίας.



Εικόνα 5 - Ψηφιακή Επεξεργασία Εικόνας

#### 2.1.2 Ιστορική Αναδρομή

Πολλές από τις τεχνικές της ψηφιακής επεξεργασίας εικόνας, αναπτύχθηκαν κατά την δεκαετία του '60 στο Jet Propulsion Laboratory, MIT, Bell Labs, του Πανεπιστημίου του Maryland και σε μερικά ακόμα μέρη, με εφαρμογή στις δορυφορικές εικόνες, πρότυπα μετατροπής τηλεγραφικής φωτογραφίας, σε ιατρική απεικόνιση, στα τηλέφωνα με χρήση βίντεο, σε αναγνώριση χαρακτήρων

και ενίσχυση (απορρόφηση;) φωτογραφιών. Αλλά και το κόστος της επεξεργασίας ήταν αρκετά υψηλό με τον εξοπλισμό των υπολογιστών εκείνης της εποχής.

Κατά την δεκαετία του '70, η ψηφιακή επεξεργασία εικόνας πολλαπλασιάστηκε καθώς φθηνότεροι υπολογιστές και εξειδικευμένο hardware έγιναν διαθέσιμα. Οι εικόνες μπορούσαν πλέον να υποβληθούν σε περαιτέρω επεξεργασία σε πραγματικό χρόνο, για κάποια συνηθισμένα προβλήματα όπως η μετατροπή τηλεοπτικών προτύπων. Δεδομένου ότι οι υπολογιστές γενικής χρήσης έγιναν γρηγορότεροι άρχισαν να αναλαμβάνουν το ρόλο του υλικού σε όλες αλλά και στις περισσότερο εξειδικευμένες και εντατικά υπολογιστικές διαδικασίες.

Κατά την δεκαετία του 2000 με γρήγορους πλέον υπολογιστές και επεξεργαστές σημάτων διαθέσιμους, η ψηφιακή επεξεργασία εικόνας αναδείχθηκε στην πιο κοινή μορφή επεξεργασίας εικόνων, και ευρύτερα χρησιμοποιούμενη όχι μόνο επειδή είναι η πιο ευπροσάρμοστη μέθοδος αλλά επίσης και η πιο οικονομική.

### **2.1.3 Στόχοι**

Η ψηφιακή επεξεργασία εικόνας επιτρέπει την χρήση πολλών περισσότερων σύνθετων αλγορίθμων για επεξεργασία εικόνας και ως εκ τούτου μπορεί να προσφέρει και τα δύο, περιπλοκότερη απόδοση στους απλούς στόχους και την εφαρμογή μεθόδων οι οποίες θα ήταν αδύνατες με τα αναλογικά μέσα. Πιο συγκεκριμένα, η ψηφιακή επεξεργασία εικόνας είναι η μοναδική πρακτική τεχνολογία για:

- Ταξινόμηση
- Εξαγωγή χαρακτηριστικών
- Αναγνώριση σχεδίων
- Προβολή
- Ανάλυση σημάτων πολυ-κλίμακας

Μερικές τεχνικές που χρησιμοποιούνται στην ψηφιακή επεξεργασία εικόνας περιλαμβάνουν:

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

[6]

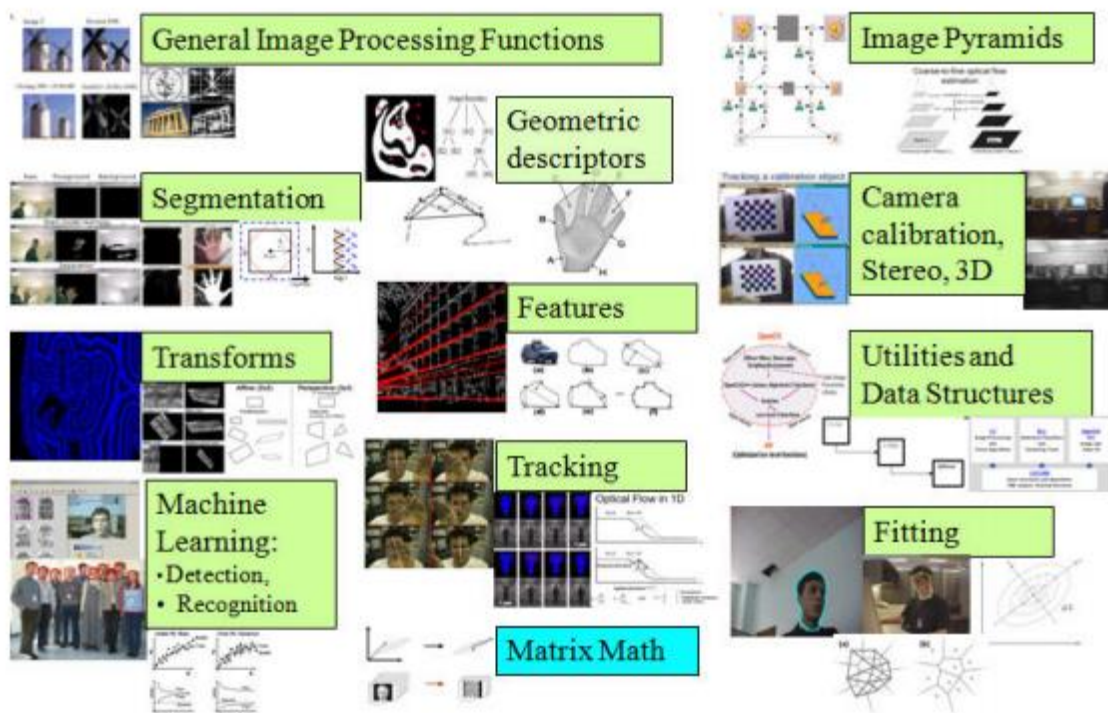
## 2.2 Βιβλιοθήκη OpenCV

Η OpenCV είναι μια βιβλιοθήκη ελεύθερου λογισμικού η οποία αναπτύχθηκε από την Intel και αφορά στην επεξεργασία εικόνας. Η OpenCV παρέχει μία μεσαίου έως υψηλού επιπέδου διασύνδεση εφαρμογών με περίπου τριακόσιες συναρτήσεις γραμμένες σε C και μερικές κλάσεις C++. Η ανάπτυξη επικοινωνίας ανθρώπου με υπολογιστή, η ανίχνευση, απομόνωση και αναγνώριση αντικειμένων, η ανίχνευση και αναγνώριση προσώπων, η κατανόηση και παρακολούθηση κίνησης είναι μερικά από τα πεδία μηχανικής όρασης που καλύπτει. Οι αλγόριθμοι της OpenCV είναι βελτιστοποιημένοι για επεξεργαστές αρχιτεκτονικής Intel Pentium (MMX, Pro, 3, 4). Τέλος, η δημιουργία της OpenCV αποσκοπεί και στην δημιουργία μιας κοινότητας ανοιχτού λογισμικού, σχετική με την μηχανική όραση, η οποία θα αναπτύσσει σύγχρονες μεθόδους επεξεργασίας εικόνας σε ένα συνεχώς αναπτυσσόμενο τεχνολογικό περιβάλλον.

Η βιβλιοθήκη αυτή είναι ανοιχτού κώδικα και είναι ελεύθερη για εκπαιδευτική αλλά και για εμπορική χρήση. Είναι γραμμένη κυρίως στην γλώσσα προγραμματισμού C και χρησιμοποιείται στις κυριότερες πλατφόρμες όπως Windows, Linux, Mac, Android και άλλα. Η OpenCV έχει ενσωματωμένους αλγόριθμους επεξεργασίας εικόνας και τεχνητής όρασης. Μερικές από τις εφαρμογές που καλύπτει η OpenCV είναι:



- Αναγνώρισης Προσώπου
- Αναγνώρισης Χειρονομιών
- Διεπαφής Χρήστη-Υπολογιστή
- Κινητά Ρομπότ
- Αναγνώρισης και Ταυτοποίησης αντικειμένων

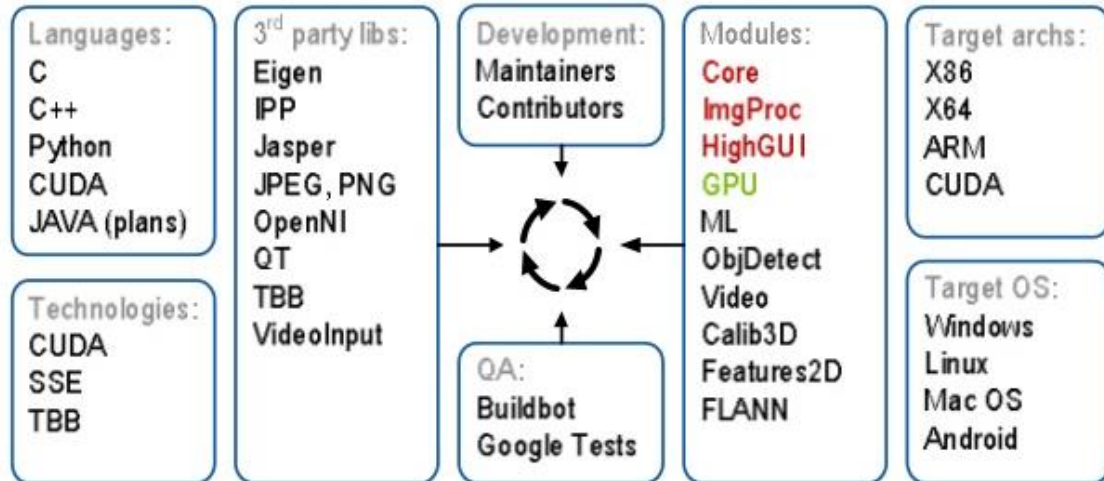


**Εικόνα 6 - Εφαρμογές OpenCV**

Στην OpenCV κάθε εικόνα αναπαριστάτε με έναν πολυδιάστατο πίνακα. Ο τύπος αυτού του πολυδιάστατου πίνακα είναι τύπου Mat και μπορεί να περιέχει αρκετές διαστάσεις πινάκων , οι οποίοι συνήθως αντιπροσωπεύουν την απεικόνιση της εικόνας σε διάφορα χρώματα (rgb) και τιμή alpha για το μοντέλο rgba ενώ υπάρχουν και άλλα μοντέλα. Ο τύπος Mat παρέχει επίσης στον προγραμματιστή εύκολη πρόσβαση σε κάθε στοιχείο του πίνακα , με μεγάλη ταχύτητα . Η βιβλιοθήκη παρέχει μία πλειάδα από συναρτήσεις , οι οποίες καλύπτουν το ευρύ φάσμα της αναγνώρισης προτύπων και την επεξεργασία εικόνας. Σε επόμενο κεφάλαιο θα δούμε αναλυτικότερα κάποιες από της συναρτήσεις που χρησιμοποιήσαμε.

Η αρχιτεκτονική του OpenCV παρουσιάζεται στην παρακάτω εικόνα:

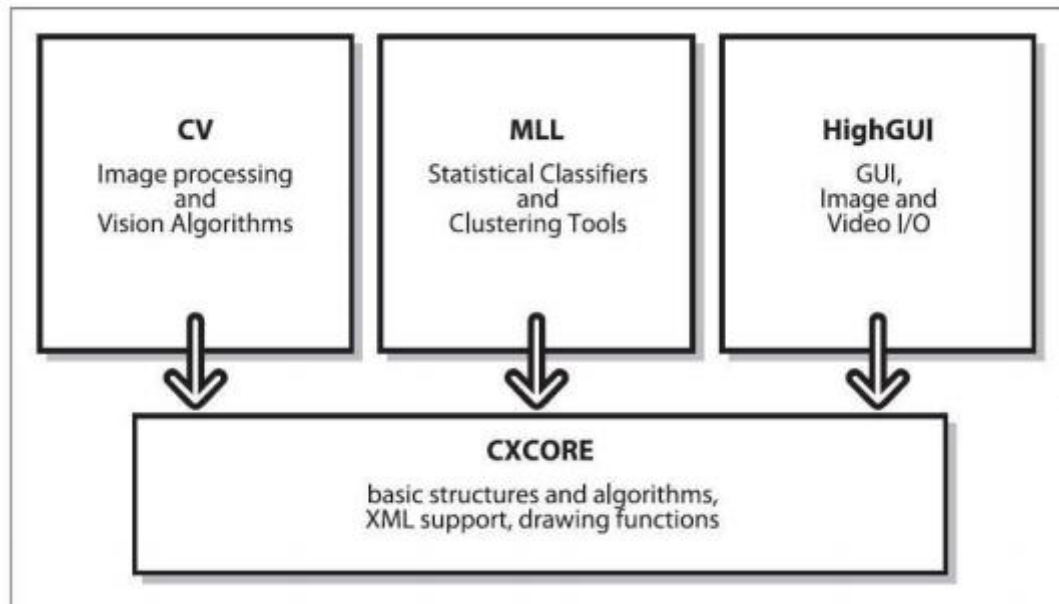
## OpenCV Architecture and Development



**Εικόνα 7 - Αρχιτεκτονική OpenCV**

Η OpenCv περιέχει πάνω από 500 συναρτήσεις, κατάλληλες για πολλές διαφορετικές εφαρμογές στους τομείς της οπτικής επεξεργασίας, όπως για ιατρικές εφαρμογές που αφορούν εικόνα και βίντεο, κλειστό κύκλωμα παρακολούθησης, ασφάλεια, ρομποτική, ρύθμιση κάμερας, διεπαφή χρήστη και υπολογιστή. Επειδή υπάρχει ισχυρή συσχέτιση μεταξύ μηχανικής όρασης και μηχανικής εκμάθησης, η OpenCv περιέχει και μια βιβλιοθήκη μηχανικής εκμάθησης γενικού σκοπού (ML - Machine Learning Library). Αυτή η ειδική βιβλιοθήκη εστιάζει στην αναγνώριση προτύπων με στατιστικές μεθόδους και στην ομαδοποίηση. Χρησιμοποιείται σε οποιοδήποτε πρόβλημα μηχανικής όρασης, λόγω της γενικής μορφής της. Η OpenCv δομείται κυρίως από 5 συστατικά μέρη, 4 από τα οποία παρουσιάζονται στην (βλ. εικόνα παρακάτω). Το πρώτο στο σχήμα με την ονομασία CV, αναπαριστά τους βασικούς αλγορίθμους για επεξεργασία εικόνας, αλλά και τους υψηλού επιπέδου για τη μηχανική όραση. Το δεύτερο, ML, αναπαριστά τη βιβλιοθήκη μηχανικής μάθησης, που περιλαμβάνει εργαλεία για ομαδοποίηση και στατιστική ταξινόμηση. Το τρίτο, HighGUI, αναπαριστά τις ρουτίνες εισόδου-εξόδου, τις συναρτήσεις για την προβολή και την αποθήκευση των βίντεο και γενικά ό,τι έχει να κάνει με τη διεπαφή χρήστη-εφαρμογής. Τέλος, το CxCore περιέχει τις βασικές δομές δεδομένων, όπως το

Imgproc, τη βασική δομή στην OpenCv που περιγράφει με λεπτομέρεια το κάθε καρέ προς επεξεργασία. Το τελευταίο δομικό μέρος της OpenCv είναι το CxCore, που περιέχει αλγορίθμους που δεν χρησιμοποιούνται πολύ, γιατί είναι σε πειραματικό στάδιο ή που έχουν απλά εγκαταλειφθεί.



Εικόνα 8 - Βασικά συστατικά της OpenCV

### 2.2.1 Τα πλεονεκτήματα της 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.

### 2.2.2 Εφαρμογές της OpenCV

Οι επιστήμονες και οι ακαδημαϊκοί, που ασχολούνται με τη μηχανική όραση, γνωρίζουν πραγματικά τις δυνατότητες της και το ευρύ φάσμα στο οποίο χρησιμοποιείται. Οι περισσότεροι γνωρίζουν κάποια από τα πλεονεκτήματα της μηχανικής όρασης και λίγες εφαρμογές της. Οι πιο γνωστές εφαρμογές της είναι στα συστήματα παρακολούθησης, σε εικόνες και βίντεο στο διαδίκτυο. Η πραγματική δύναμή της φαίνεται σε περιοχές που δεν είναι ευρέως γνωστό ότι χρησιμοποιείται, όπως σε διεπαφές παιχνιδιών και σε άλλες εντυπωσιακές, όπως στους περισσότερους εναέριους χάρτες και χάρτες δρόμων (όπως στο Google Street View), όπου χρησιμοποιούν κατά κόρον τεχνικές συγκόλλησης εικόνων, ευθυγράμμισης και διόρθωσης της εικόνας από παραμορφώσεις που εισάγει ο φακός. Άλλες χρήσεις της είναι σε μη-επανδρωμένα οχήματα (γενικά στη ρομποτική) ή βιοϊατρικές αναλύσεις. Ένας σημαντικός κλάδος στον οποίο κερδίζει έδαφος, είναι η βιομηχανία. Σχεδόν όλα τα προϊόντα που παράγονται μαζικά, έχουν επιθεωρηθεί σε κάποιο στάδιο της παραγωγής από συστήματα μηχανικής όρασης. Από την πρώτη της έκδοση, τον Ιανουάριο του 1999, η OpenCv έχει χρησιμοποιηθεί σε πολλές εφαρμογές, προϊόντα και ερευνητικές μελέτες.

Αυτές οι εφαρμογές περιλαμβάνουν τον συγκερασμό εικόνων σε δορυφορικούς και διαδικτυακούς χάρτες, τη μείωση του θορύβου σε ιατρικές εικόνες, τα αυτόματα συστήματα ελέγχου και ασφαλείας, στρατιωτικές εφαρμογές και μη-επανδρωμένα οχήματα (εναέρια, υποβρύχια, επίγεια). Έχει χρησιμοποιηθεί, ακόμα, σε αναγνώριση ήχου και μουσικής. Η OpenCv ήταν ένα βασικό συστατικό του συστήματος όρασης του ρομπότ από το πανεπιστήμιο Stanford, του "Stanley", το οποίο κέρδισε τον DARPA Grand Challenge, αγώνα δρόμου ρομπότ στην έρημο.

### 2.2.3 Βασικές Δομές & 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

- CvtColor: 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 καρέ 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 BGR 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 ενός αντικειμένου. Μόλις κληθεί η συνάρτηση αυτή όλες οι ροπές αποθηκεύονται στο διάνυσμα m.

[7]

## 2.3 Επεξεργασία Εικόνας

### 2.3.1 Επεξεργασία με Γλώσσα Προγραμματισμού Python

Επέλεξα σαν εργαλείο για την υλοποίηση της πτυχιακής μου να χρησιμοποιήσω αν και όχι ευρέως διαδεδομένη στην Ελλάδα (ας ελπίσουμε όχι για πολύ) την υψηλού επιπέδου γλώσσα προγραμματισμού Python. Η Python είναι μια αλληλεπιδραστική γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Ολλανδό Guido van Rossum το 1990. Αρχικά ήταν scripting γλώσσα για το λειτουργικό σύστημα Amoeba και ικανή για κλήσεις συστήματος. Η γλώσσα χρησιμοποιεί μεταγλωττιστή για την δημιουργία του εκτελέσιμου κώδικα και συσχετίζεται με τις γλώσσες προγραμματισμού Tcl, Perl, Scheme, Java και Ruby. Η Python έχει αναπτυχθεί ως ανοιχτό λογιστικό (ανοιχτού κώδικα) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Στο ερώτημα γιατί Python έχω να απαντήσω τα εξής :

- Μία από τις πιο ανεπτυγμένες εταιρίες υψηλής τεχνολογία που πρωτοπορεί σε καινοτόμα προϊόντα σήμερα, η πασίγνωστη σε όλους μας Google χρησιμοποιεί την Python από την αρχή της ανάπτυξής της :

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves" -- Peter Norvig, Google.

- Είναι μία γλώσσα που λόγω της απλότητάς της και δυναμικότητας του scripting language ενθαρρύνεται για rapid development.

- Η σύνταξή της είναι τόσο απλή σαν ψευδοκώδικα, όταν ο κώδικας που βλέπουμε δείχνει λογικά σωστός , συνήθως το πρόγραμμα θα τρέχει σωστά.

- Προσφέρει μεγάλη υποστήριξη για ενοποίηση με άλλες γλώσσες προγραμματισμού και εργαλεία.

- Διανέμετε με άδεια OSI-approved που σημαίνει ότι είναι μία γλώσσα 100% open source και οτιδήποτε δημιουργήσουμε με αυτήν μπορούμε να το χρησιμοποιήσουμε ελεύθερα χωρίς κανέναν περιορισμό, ακόμα και ως μία εμπορική εφαρμογή.

- Είναι 100% portable, ο ίδιος κώδικας τρέχει σε Windows, Linux, Mac κτλ. χωρίς μετατροπές.
- Μόλις πρόσφατα η Microsoft έκανε τα πρώτα βήματα για να την ενσωμάτωση στην οικογένεια της .NET με την κωδική ονομασία IronPython.

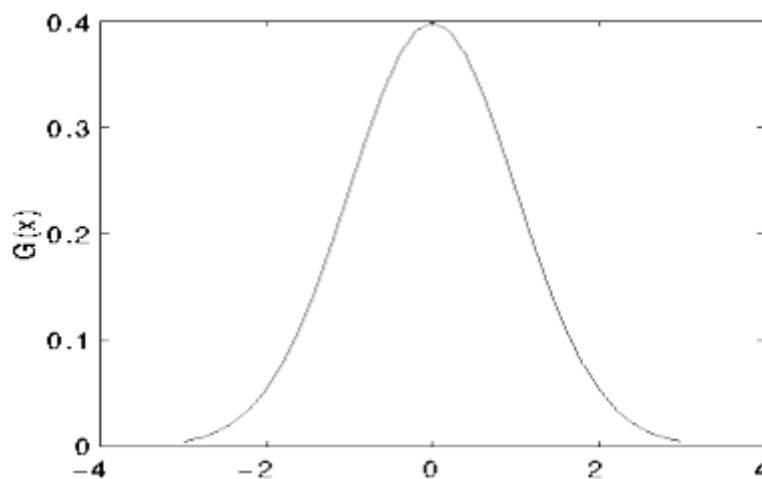
[8]

### 2.3.2 Φίλτρο Gaussian

Το φίλτρο αυτό χρησιμοποιείται για το θόλωμα της εικόνας και την ελάττωση του θορύβου. Οι ομαλές περιοχές της εικόνας, χωρίς μεγάλες μεταβολές, αντιστοιχούν σε χαμηλές συχνότητες. Αντίθετα οι περιοχές που παρουσιάζουν μεγάλες μεταβολές (π.χ. ακμές, ή όρια διαφορετικών περιοχών της εικόνας) αντιστοιχούν σε ψηλές συχνότητες. Η Gaussian κατανομή σε 1-D έχει τη μορφή:

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

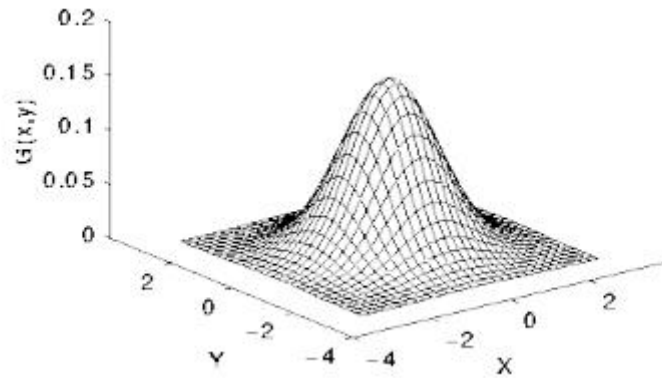
όπου  $\sigma$  είναι η τυπική απόκλιση της κατανομής. Έχουμε επίσης υποθέσει ότι η κατανομή έχει μέσο μηδέν (δηλαδή είναι επικεντρωμένη στη γραμμή  $x = 0$ ). Η κατανομή φαίνεται στην παρακάτω εικόνα.



Εικόνα 9 - Gaussian κατανομή 1-D

Σε 2-D, μια ισότροπη (δηλαδή κυκλικά συμμετρική) Gaussian έχει τη μορφή:

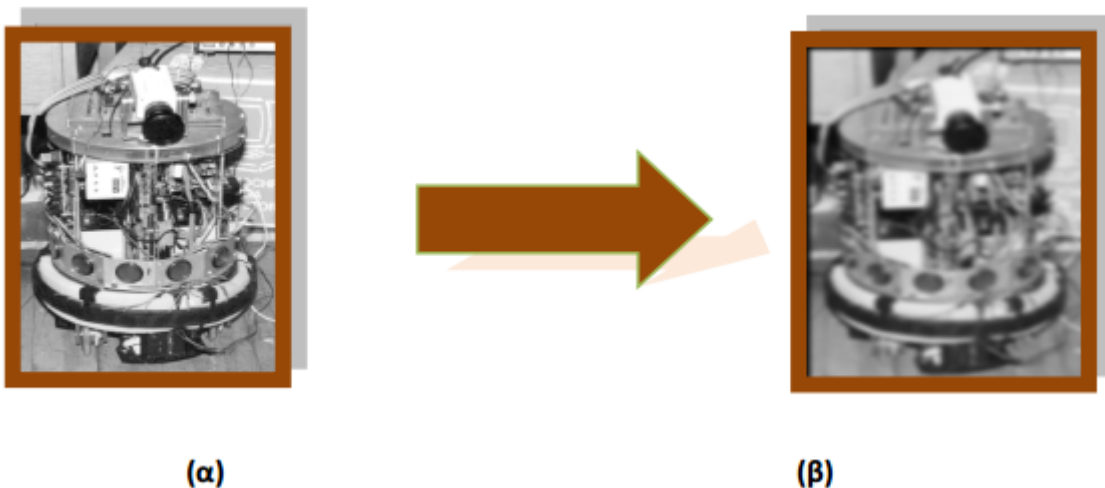
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Εικόνα 10 - Gaussian κατανομή 2-D

Η ιδέα της Gaussian εξομάλυνσης επιτυγχάνεται με συνέλιξη. Δεδομένου ότι η εικόνα αποθηκεύεται ως μια συλλογή από διακριτά εικονοστοιχεία πρέπει να παράγουμε μία διακριτή προσέγγιση της Gaussian λειτουργίας. Θεωρητικά, η Gaussian κατανομή είναι μη μηδενική παντού.

[9]

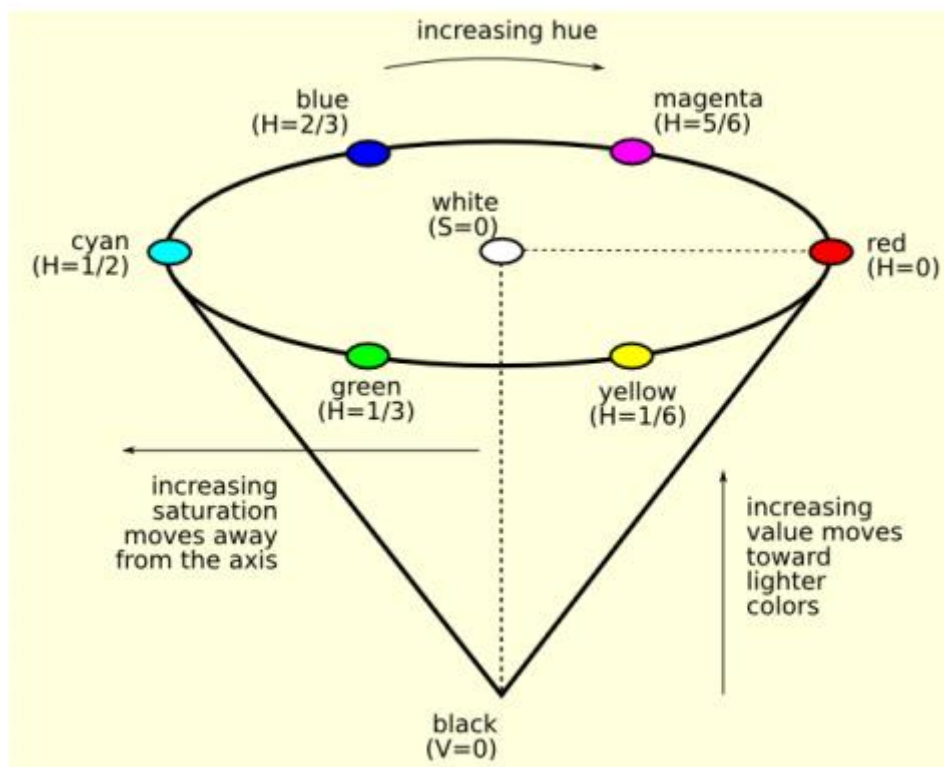


Εικόνα 11 - (α) Κανονική εικόνα, (β) Εικόνα με φιλτράρισμα



### 2.3.3 Χρωματικό Μοντέλο HSV

Το χρωματικό μοντέλο HSV επινοήθηκε το 1978 από τον A. R. Smith. Το σύστημα συντεταγμένων είναι κυκλικό και τα χρώματα βρίσκονται μέσα σε ένα εξαγώνο κώνο. Οι κύριες παράμετροι που χρησιμοποιεί : Η απόχρωση (Hue) αντιπροσωπεύει τον τύπο του 'καθαρού' χρώματος. Για παράδειγμα όλες οι αποχρώσεις και οι τόνοι του κόκκινου χρώματος έχουν την ίδια τιμή Hue. Οι αποχρώσεις περιγράφονται από ένα αριθμό που προσδιορίζει τη θέση του αντίστοιχου καθαρού χρώματος στον χρωματικό τροχό ως ένα κλάσμα μεταξύ μηδέν και ένα. Η τιμή Hue μηδέν αναφέρεται στο κόκκινο, η τιμή Hue 1/6 αναφέρεται στο κίτρινο, η τιμή Hue 1/3 στο πράσινο κ.ο.κ. Η χρωματική καθαρότητα (Saturation) είναι ο βαθμός μείξης ενός καθαρού χρώματος με το άσπρο. Το καθαρό κόκκινο είναι πλήρως καθαρό με Saturation 1, οι αποχρώσεις του έχουν Saturation μικρότερο από 1 και το άσπρο έχει Saturation 0. Η τιμή (Value) αναφέρεται στο βαθμό μείξης ενός καθαρού χρώματος με το μαύρο, είναι δηλαδή παράμετρος φωτεινότητας. Το μαύρο έχει Value 0 και η φωτεινότητα των χρωμάτων αυξάνεται καθώς απομακρύνονται από το μαύρο.



Εικόνα 12 - Χρωματικός χώρος HSV

Το εξωτερικό άκρο της κορυφής του κώνου είναι ο τροχός χρώματος με όλα τα καθαρά χρώματα. Η παράμετρος της απόχρωσης περιγράφει τη γωνία γύρω από τον τροχό. Ακόμα, η παράμετρος της χρωματικής καθαρότητας είναι μηδέν για κάθε χρώμα επί του άξονα του κώνου και το κέντρο του κύκλου στην κορυφή είναι άσπρο. Μια αύξηση στην τιμή της αντιστοιχεί σε μια κίνηση μακριά από τον άξονα. Τέλος, η τιμή (Value) είναι μηδέν για το μαύρο. Μια αύξηση στην τιμή της αντιστοιχεί σε μια κίνηση μακριά από το μαύρο προς την κορυφή του κώνου. Οι κάμερες του στερεοσκοπικού συστήματος χρησιμοποιούν το RGB μοντέλο, το οποίο στην OpenCV αναφέρεται ως BGR. Μόλις η αριστερή κάμερα διαβάσει αυτές τις τιμές, τις μετατρέπει σε HSV τιμές. Στη συνέχεια χρησιμοποιούνται στον κώδικα για να καθορίσουν την θέση του συγκεκριμένου αντικειμένου το οποίο πρέπει να αναγνωρίσει ο ρομποτικός βραχίονας. Τα εικονοστοιχεία ελέγχονται έτσι ώστε να αποφασιστεί αν ταιριάζουν με ένα προκαθορισμένο χρωματικό κατώφλι.

```
//convert frame from BGR to HSV colorspace

cvtColor(temp,HSV,COLOR_BGR2HSV);

//filter HSV image between values and store filtered image to threshold matrix
if(red==1){
    inRange(HSV,Scalar(150,0.51*255,0.51*255),Scalar(179,255,255),threshold);
}
if(blue==1){
    inRange(HSV,Scalar(92,0.780*255,0.82*255),Scalar(150,255,255),threshold);
}
if(green==1){
    inRange(HSV,Scalar(60,0.457*255,0.567*255),Scalar(90,255,255),threshold);
}
if(yellow==1){
    inRange(HSV,Scalar(20,0.448*255,0.573*255),Scalar(55,255,255),threshold);
}
if(orange==1){
    inRange(HSV,Scalar(2,0.488*255,0.706*255),Scalar(10,255,255),threshold);
}
}
```

Εικόνα 13 - Μετατροπή εικόνας σε χρωματικό χώρο HSV

Στην OpenCV ο χρωματικός χώρος HSV αποτελείται επίσης από τρεις πίνακες. Οι πίνακες H, S, V παίρνουν τιμές 0 – 179, 0 – 255, 0 – 255 αντίστοιχα. Η OpenCV έχει ενσωματωμένη την εντολή `cvtColor()` η οποία δέχεται ως είσοδο την αρχική εικόνα (`source image`) σε BGR και δίνει στην έξοδο την εικόνα (`destination image`) μετασχηματισμένη σε άλλο χρωματικό χώρο. Επίσης, ορίζεται μια παράμετρος που δείχνει ποιος χρωματικός μετασχηματισμός θα γίνει. Στη συγκεκριμένη περίπτωση χρησιμοποιείται η παράμετρος `COLOR_BGR2HSV` για μετατροπή στον HSV χώρο. Στη συνέχεια καλείται η συνάρτηση `inRange()`, που δέχεται την HSV εικόνα που έδωσε η `cvtColor()` και δίνει στην έξοδο την εικόνα “κατώφλι” σε δυαδική μορφή (`binary`), δηλαδή θα περιέχει μόνο άσπρα και μαύρα στοιχεία. Το μέγεθος της κατωφλίωσης ορίζεται από το κάτω και πάνω όριο (`low boundary`, `upper boundary` αντίστοιχα), τα οποία είναι τύπου `Scalar(x,y,z)`. Πιο συγκεκριμένα, στην περίπτωση όπου οι τιμές H, S, V των εικονοστοιχείων της εικόνας εισόδου είναι χαμηλότερες από τα x, y, z του κάτω ορίου τίθενται “0” (μαύρα) στην εικόνα κατωφλίου. Αντίστοιχα, αν οι τιμές H, S, V των εικονοστοιχείων είναι μεγαλύτερες ή ίσες από τα x, y, z του πάνω ορίου τίθενται “0” (μαύρα) στην εικόνα κατωφλίου. Επομένως, τα εικονοστοιχεία με τιμές μεταξύ των ορίων παίρνουν τιμή “255” (άσπρα) και ουσιαστικά αποτελούν το αντικείμενο που πρέπει να αναγνωριστεί.

[10]

### 2.3.4 Επεξεργασία Εικόνας και Εύρεση Γραμμής

Σε αυτή την ενότητα θα αναλύσουμε την διαδικασία επεξεργασίας της εικόνας για την εύρεση της γραμμής (`Line detect`). Παρακάτω θα περιγραφεί ένας βασικός αλγόριθμος ψηφιακής ανάλυσης εικόνας όπου αφού επεξεργαστεί η εικόνα, θα βρίσκεται η γωνία κλίσης της χρωματιστής κόκκινης γραμμής. Τα βήματα τα οποία εκτελεί ο αλγόριθμος είναι τα παρακάτω:

- Ξεκινάει η κάμερα να τραβάει βίντεο
- Διαβάζουμε την εικόνα καρέ-καρέ
- Χρησιμοποιούμε το φίλτρο `Gaussian` στη εικόνα
- Αλλάζουμε το χρωματικό χώρο της εικόνας
- Βρίσκουμε το μέσο της κόκκινης γραμμής (Συντεταγμένες x,y)
- Συγκρίνουμε τις συντεταγμένες x, y με τις προηγούμενες `xOld`, `yOld`
- Ανάλογα με την σύγκριση το όχημα στρίβει δεξιά, αριστερά ή

πηγαίνει ευθεία

Ο αλγόριθμος προϋποθέτει ότι δεν υπάρχουν πολλές αλλαγές στα χρώματα στην εικόνα που επεξεργάζεται. Για έναν δρόμο και μία κόκκινη γραμμή στη μέση θεωρώ ότι είναι κατάλληλος.

Παρακάτω θα αναλύσουμε τα κυριότερα κομμάτια κώδικα του βασικού αλγορίθμου της εργασία με όνομα linedetect.py, όπου υπάρχει και στο παράρτημα:

- Στις παρακάτω εντολές, το πρόγραμμα ξεκινάει να τραβάει βίντεο ζωντανά (live streaming) και ρυθμίζουμε τις παραμέτρους του, δηλαδή το πλάτος και το ύψος του βίντεο.

```
cap = cv2.VideoCapture(0)
```

```
cap.set(3,600)
```

```
cap.set(4,400)
```

- Σε αυτό το βήμα, το πρόγραμμα διαβάζει την εικόνα του βίντεο καρέ-καρέ, θολώνουμε την εικόνα με το φίλτρο Gaussian για να περιορίσουμε τον θόρυβο της εικόνας και αλλάζουμε το χρωματικό χώρο της εικόνας σε HSV.

```
ret, image = cap.read()
```

```
blur = cv2.GaussianBlur(image, (5,5), 0)
```

```
hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)
```

- Παρακάτω ορίζουμε τα όρια του κόκκινου χρώματος ως RGB με τιμές από ανοιχτό μέχρι σκούρο κόκκινο. Μετά το πρόγραμμα ελέγχει αν υπάρχει στην εικόνα κόκκινο χρώμα και αν υπάρχει το επιστρέφει στην μεταβλητή mask, και στη συνέχεια εφαρμόζουμε το φίλτρο στην μεταβλητή mask όπου υπάρχει η κόκκινη γραμμή.

```
lower_red = np.array([0,100,100])
upper_red = np.array([20,255,255])
mask = cv2.inRange(hsv, lower_red, upper_red)
bmask = cv2.GaussianBlur(mask, (5,5), 0)
```

- Το πρόγραμμα σε αυτό το βήμα με την εντολή `cv2.moments` κάνει αποσυμπίεση των στιγμών της κόκκινης γραμμής και τις αποθηκεύει σε έναν πίνακα με τιμές { `m00,m01,m10,m11...`}. Με αυτόν τον τρόπο βρίσκουμε τις συντεταγμένες `x, y` όπου είναι και το κέντρο της γραμμής.

```
moments =cv2.moments(bmask)
m00 = moments['m00']
cx = int(moments['m10']/m00)
cy = int(moments['m01']/m00)
```

- Στις παρακάτω εντολές το πρόγραμμα δημιουργεί ένα παράθυρο όπου δείχνει στην οθόνη την εικόνα που τραβάει η κάμερα καθώς επίσης και έναν κύκλο όπου μας δείχνει το κέντρο της γραμμής.

```
cv2.imshow('frame', image)
cv2.circle(image,ctr,10,(10,10,10))
```

- Παρακάτω το πρόγραμμα με αυτές τις εντολές τυπώνει τις συντεταγμένες `x, y` της εικόνας καθώς και τις προηγούμενες `xOld, yOld`. Συγκρίνουμε τις παλιές με τις καινούργιες συντεταγμένες κυρίως στον άξονα `x` γιατί εκεί υπάρχει μεταβολή και ανάλογα με την σύγκριση το όχημα θα στρίψει αριστερά ή δεξιά ή θα πάει ευθεία.

```
print cx, cy
print cxOld, cyOld
```

```
if (cxOld-cx) > 10:
```

```
    print "move left"
```

```
    left()
```

```
    sleep(0.1)
```

```
    go()
```

```
if (cx-cxOld) > 10:
```

```
    print "move right"
```

```
    right()
```

```
    sleep(0.1)
```

```
    go()
```

```
if (cxOld-cx < 10 and cx-cxOld < 10):
```

```
    print "go"
```

```
    go()
```



## ΚΕΦΑΛΑΙΟ 3

### 3.1 Υλικά Κατασκευής

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

- Raspberry Pi 3 πλακέτα
- Usb Κάμερα (camera module)
- Micro SD card
- Αυτοκινητάκι που λειτουργεί με ηλεκτρικούς κινητήρες (dc motors)
- L293D ολοκληρωμένο (chip)
- Μικρή πλακέτα (breadboard)
- Θηλυκά και αρσενικά καλώδια (jumpers)
- 2 Εξωτερικούς φορτιστές μπαταριών(powerbank)
- Εξωτερική θήκη raspberry pi
- Πληκτρολόγιο
- Ποντίκι

Ενδεικτικά παραθέτω παρακάτω φωτογραφίες και σύντομες περιγραφές με τα υλικά που χρησιμοποιήθηκαν.

#### 3.1.1 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 τις κάνει ένας προσωπικός υπολογιστής, όπως προβολή ταινιών, παιχνίδια, πλοήγηση στο διαδίκτυο κλπ.



Λόγω του μικρού μεγέθους και των υψηλών του δυνατοτήτων του χρησιμοποιείται σε πολλά προγράμματα (projects). Ο λόγος επιλογής αυτής της πλακέτας (board) ήταν ότι αποτελεί ένα σύγχρονο υπολογιστικό σύστημα με πολλές λειτουργικότητες και χαμηλό κόστος. Έχουν κυκλοφορήσει τρία μοντέλα, το raspberry pi 1, το raspberry pi 2 (Model B & Model B+) ενώ πρόσφατα κυκλοφόρησε και το raspberry pi 3.

Η διαφορά του μοντέλου raspberry pi 3 με το αρχικό raspberry pi 1 είναι ότι διαθέτει θύρα Ethernet, παραπάνω θύρες USB 2.0 και περισσότερη μνήμη Ram. Για την διπλωματική εργασία επιλέχθηκε το Raspberry Pi 3.

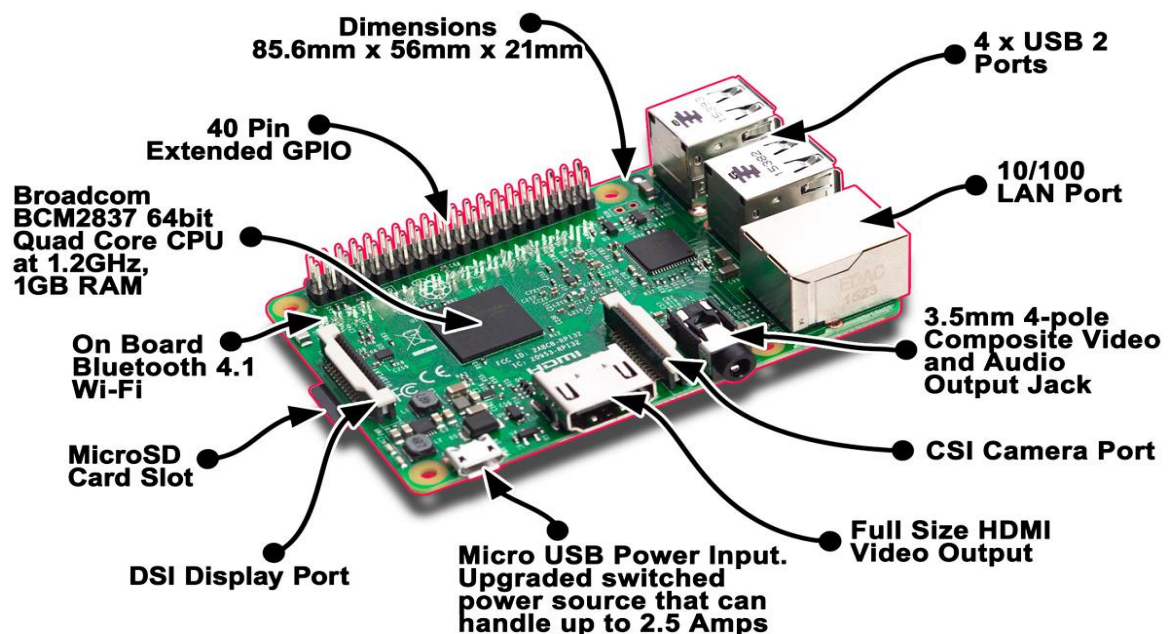
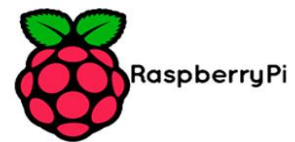
Το raspberry pi διαθέτει μία SD Card η οποία χρησιμοποιείται για την αποθήκευση αρχείων. Επίσης η SD Card είναι απαραίτητη στην boot διαδικασία καθώς όλα τα απαραίτητα αρχεία για την διαδικασία αυτή αποθηκεύονται σε ένα FAT32 partition της κάρτας.

Όπως μπορεί να παρατηρήσει κανείς στο Raspberry Pi, υπάρχει ένας Header ο οποίος διαθέτει 40 ακίδες. Πρόκειται για τον P1 Header. Συνολικά στη πλακέτα υπάρχουν ακόμη πέντε headers αλλά δεν είναι ενεργοποιημένοι.

Ο P1 header είναι γνωστός και ως expansion header 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 Pi επιτυγχάνεται μέσω μιας micro usb θύρας που υπάρχει στη πλακέτα ή μέσω των 5V και GND ακίδων(pins).

Τα χαρακτηριστικά του Raspberry Pi 3 είναι τα εξής:

- 4 θύρες USB (μπορούν να χρησιμοποιηθούν για ποντίκι και πληκτρολόγιο)
- 1 θύρα ethernet (για σύνδεση στο τοπικό δίκτυο)
- 1 κάρτα SD (παίζει το ρόλο του αποθηκευτικού μέσου, όχι μόνο για τα αρχεία σας αλλά και για το λειτουργικό που θα τρέξει)
- 1 θύρα HDMI (για την σύνδεση με την οθόνη)
- 1 αναλογική έξοδος (για τα ηχεία)
- 1 rca video έξοδος (σε περίπτωση που δεν υπάρχει οθόνη με HDMI)
- 1 micro usb (για την τροφοδοσία)
- Ενσωματωμένο Wifi και Bluetooth



Εικόνα 14 - Raspberry Pi 3

Θα αναλύσουμε τα χαρακτηριστικά του Raspberry Pi 3 και κάποιες διαφορές που υπάρχουν σε σχέση με πιο παλιά μοντέλα. Το πρώτο χαρακτηριστικό είναι ότι έχει αλλάξει ο επεξεργαστής και είναι ο BCM2837 ο οποίος αποτελείται από τέσσερις πυρήνες και είναι στα 1,2GHz. Επιπλέον η cache του επεξεργαστή αναβαθμίστηκε. Οπότε αυτόματα δίνει αρκετή δύναμη έτσι ώστε να βελτιωθούν λειτουργίες όπως η περιήγηση στο διαδίκτυο και να υπάρχουν πολλές παράλληλες λειτουργίες (multithread) όπως για παράδειγμα σε άλλον επεξεργαστή να τρέχει το γραφικό περιβάλλον και σε άλλον η επεξεργασία των δεδομένων από την sd κάρτα. Επίσης μεγάλωσε η μνήμη RAM η οποία είναι πλέον 1GB. Σύμφωνα με τον Eben Upton η ταχύτητα έχει εξαπλασιαστεί.

Μια ακόμη σημαντική διαφορά είναι ότι η αρχιτεκτονική του νέου επεξεργαστή είναι ARMv8 η οποία επιτρέπει να δημιουργηθούν διανομές λειτουργικού συστήματος όπως ubuntu και όλα τα λειτουργικά που μπορεί να έχει ένας επεξεργαστής ARM. Επίσης αίσθηση έχει προκαλέσει ότι το Raspberry Pi 3 θα είναι συμβατό με τα Windows 10.

Όσον αφορά τη συμβατότητα, οι είσοδοι-έξοδοι γενικού σκοπού παραμένουν οι ίδιες. Επίσης έχει 4 USB οι οποίες όμως παρέχουν περισσότερο ρεύμα και είναι δυνατόν να συνδεθούν συσκευές που έχουν μεγαλύτερες απαιτήσεις σε ρεύμα.

Ειδικά για το Raspberry Pi έχει φτιαχτεί και μία κάμερα, η Raspberry Pi Camera Module η οποία βοηθάει σημαντικά έτσι ώστε να τραβάμε φωτογραφίες ή βίντεο.

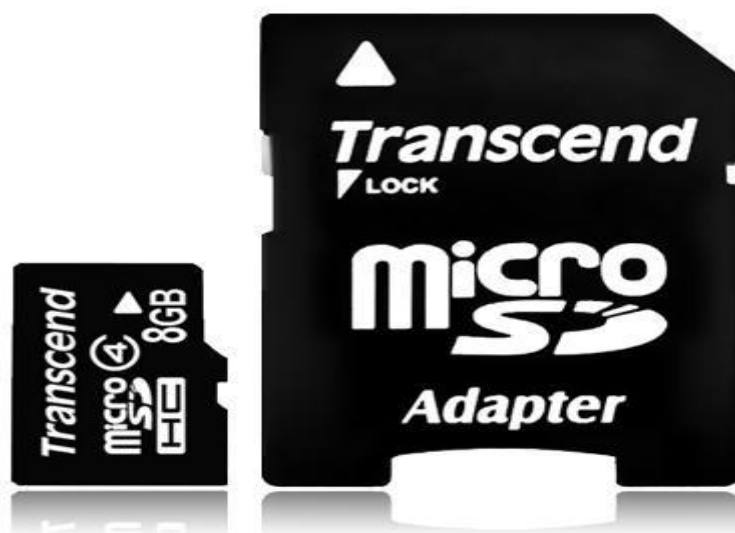
[11]

### 3.1.2 Usb Κάμερα



Εικόνα 15 - Usb Κάμερα

### 3.1.3 Micro SD Card



Εικόνα 16 - Micro SD card

### 3.1.4 Ηλεκτρικός Κινητήρας

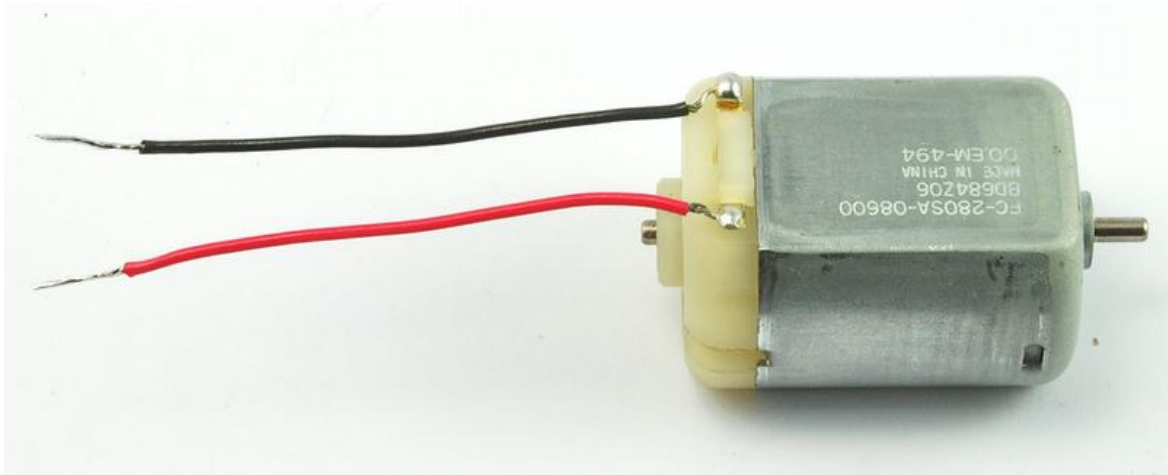
Ο ηλεκτρικός κινητήρας είναι η συσκευή που τελικά αναλαμβάνει την μετατροπή της ηλεκτρικής ενέργειας σε μηχανική και αντίστροφα. Η επιλογή του είναι καθοριστικής σημασίας για την απόδοση του ηλεκτρικού συστήματος και συνεπώς θα πρέπει να βασίζεται σε αυστηρά κριτήρια που περιλαμβάνουν την υψηλή απόδοση, 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 μόνιμων μαγνητών: Permanent Magnet Synchronous Machine (PMSM)
  - Ο κινητήρας μαγνητικής αντίδρασης: Switched Reluctance Motor (SRM)
- Στην συνέχεια παρατίθεται μια σύντομη περιγραφή των πιο διαδεδομένων κινητήρων για εφαρμογές ηλεκτρικών οχημάτων.

#### **Κινητήρας συνεχούς ρεύματος (DC Motor)**

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 ορθογώνιας θέσης του πεδίου του τυμπάνου με αυτό της διέγερσης απαιτούν απλό έλεγχο, ενώ η χρήση των μόνιμων 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 17 - DC Motor**

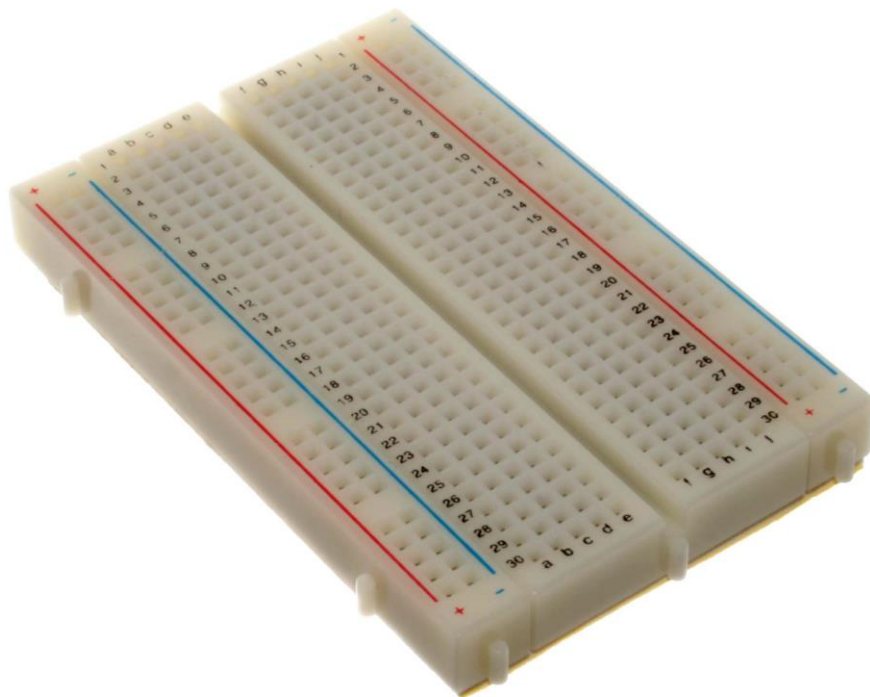
### **3.1.5 Breadboard**

Το breadboard είναι μια μονάδα για την κατασκευή προσωρινών κυκλώματα (πρωτοτύπων) και δεν απαιτούν απολύτως καμία συγκόλληση.

Προτυποποίηση (prototyping) είναι η διαδικασία ελέγχου μιας ιδέας, με τη δημιουργία ενός προκαταρκτικού μοντέλου ενός κυκλώματος. Αυτή είναι μία από τις πιο κοινές χρήσεις για breadboards. Αν δεν είστε σίγουροι για το πώς ένα κύκλωμα θα αντιδράσει κάτω από ένα δεδομένο σύνολο παραμέτρων, είναι καλύτερο να δημιουργήσετε ένα πρωτότυπο και να το δοκιμάσετε.

Για τα αρχάριους με τα ηλεκτρονικά και τα κυκλώματα, τα breadboards είναι συχνά το καλύτερο μέρος για να ξεκινήσουν . Αυτή είναι η πραγματική ομορφιά των breadboards, ότι μπορούν να στεγάσουν τόσο το απλούστερο κύκλωμα , όσο και πολύ πολύπλοκα κυκλώματα . Αν το κύκλωμά σας ξεπερνά σε μέγεθος το τρέχον breadboard, μπορεί να επεκτείνετε το κύκλωμά σας συνδέοντας άλλα breadboard για να φιλοξενήσουν κυκλώματα όλων των μεγεθών και πολυπλοκότητας.

Μια άλλη κοινή χρήση των breadboard είναι να δοκιμάσουμε νέα εξαρτήματα, όπως ολοκληρωμένα κυκλώματα (ICs). Όταν προσπαθείτε να καταλάβετε πώς ένα εξάρτημα λειτουργεί και συνεχώς αλλάζετε την καλωδίωση, δεν θέλετε να κολλάτε τις συνδέσεις σας κάθε φορά.



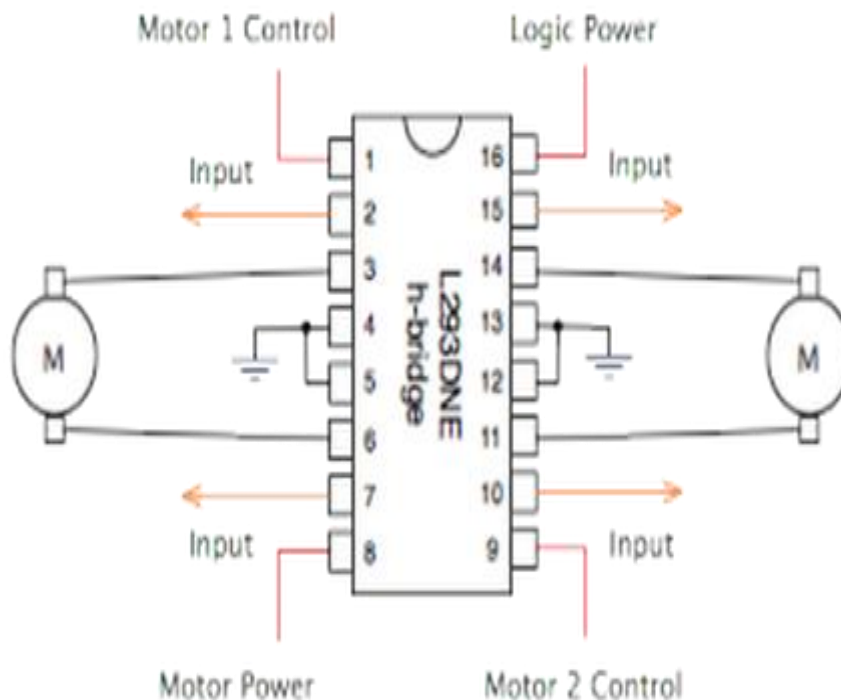
**Εικόνα 18 - Breadboard**

### 3.1.6 Ολοκληρωμένο L293D

Το ολοκληρωμένο L293D το χρησιμοποιούμε για να "οδηγήσουμε" σωστά τα μοτέρ μας. Τι σημαίνει αυτό; Είναι απλό, τα pin του raspberry pi 3 δεν μπορούν να δώσουν το ρεύμα που χρειάζεται για να κινηθούν τα μοτέρ, για αυτό τον λόγο πρέπει να χρησιμοποιήσουμε τρανζίστορ. Το L293D κάνει ακριβώς αυτή την δουλειά κρύβοντας στο εσωτερικό του 4 τρανζίστορ.

Το L293D εκτός ότι μπορεί να παρέχει το απαιτούμενο ρεύμα στα μοτέρ για να οδηγηθούν σωστά, μπορεί να δώσει και τάση μέχρι 35V, αυξάνοντας έτσι την ισχύ  $P=V \cdot I$ . Όμως, αυτό δεν σημαίνει ότι τα μοτέρ θα μπορέσουν να αποδώσουν σε τέτοιες τιμές... πιθανόν να καούν ή να καεί και το ίδιο το ολοκληρωμένο. Γι αυτό θυμηθείτε στο Pin 8 του L293D "Motor Power" να δώσετε από 5V μέχρι το πολύ 12V(!).

Το L293D δεν είναι και ότι καλύτερο υπάρχει για την οδήγηση κινητήρων DC καθώς θερμαίνεται όσο ανεβαίνει η ισχύς και μειώνετε η απόδοση του και δεν μπορεί να δώσει ρεύμα πάνω από 0.5A σε κάθε κανάλι. Ωστόσο είναι φθηνό και έχει εύκολη συνδεσμολογία και το ρεύμα του επαρκεί για την κατασκευή μας.



Εικόνα 19 - L293D chip



## 3.2 Λογισμικό

### 3.2.1 Λειτουργικό Σύστημα

Υπάρχουν πολλά διαθέσιμα λειτουργικά συστήματα για το Raspberry Pi όπως είναι το Arch Linux ARM, το Pidora (Fedora Remix), το Raspbmc και το Openelec, το Risc Os καθώς και το Raspbian που αποτελεί και την προτεινόμενη έκδοση για εφαρμογές γενικής χρήσης. Το Raspberry Pi 3 μπορεί να έχει και τα Windows 10 σαν λειτουργικό σύστημα.

Όλα τα παραπάνω είναι παραλλαγές των γνωστών λειτουργικών που τρέχουν και σε υπολογιστές που έχουν βελτιστοποιηθεί και προσαρμοστεί για τις ανάγκες του Raspberry Pi.



Εικόνα 20 - Pidora

Το Pidora είναι μια διανομή Linux για το Raspberry Pi. Περιέχει τα πακέτα λογισμικού από το λειτουργικό Fedora (συγκεκριμένα, το Fedora ARM δευτεροβάθμιας έργου αρχιτεκτονικής) που έχουν μεταγλωττιστεί ειδικά για την αρχιτεκτονική ARMv6 που χρησιμοποιείται στο Raspberry Pi, πακέτα τα οποία έχουν γραφτεί ειδικά ή έχουν τροποποιηθεί για το Raspberry Pi, και το λογισμικό που παρέχεται από την Raspberry Pi Foundation για την πρόσβαση στη συσκευή.



**Εικόνα 21 - Raspbmc**

Το Raspbmc είναι μια διανομή Linux βασισμένη στο Debian που φέρνει το XBMC στο Raspberry Pi. Το Raspberry Pi έχει αρκετή δύναμη για να χειριστεί την αναπαραγωγή πολυμέσων, καθιστώντας το ιδανικό συστατικό για την δημιουργία ενός HTPC (Home Theater PC), που μπορούσε να αποτελείται από πολύ πιο δαπανηρές πλατφόρμες που θα παρήγαγαν το ίδιο αποτέλεσμα.

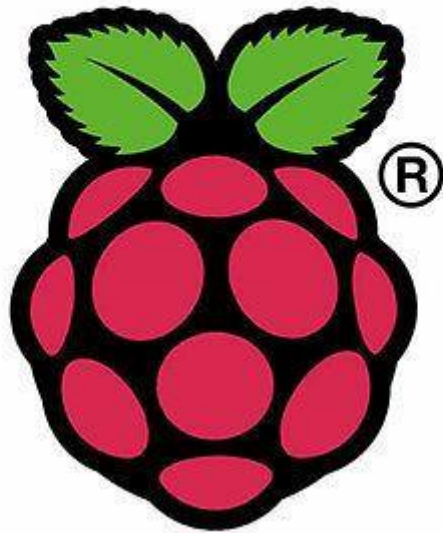


**Εικόνα 22 - Raspbian**

Το Raspbian είναι ένα ελεύθερο λειτουργικό σύστημα βασισμένο στο Debian και βελτιστοποιημένο για το υλικό του Raspberry Pi. Αποτελεί το ενδεδειγμένο λειτουργικό όταν το Raspberry Pi πρόκειται να χρησιμοποιηθεί για γενική χρήση. Έχει περίπου 35.000 πακέτα προ-μεταγλωττισμένο λογισμικό σε κατανοητή μορφή για εύκολη εγκατάσταση.

# “NOOBS”

## New Out Of Box Software



Εικόνα 23 – Noobs

Η προτεινόμενη εγκατάσταση για αρχάριους από την Raspberry Pi Foundation είναι το Noobs γίνεται με απλή αντιγραφή στην κάρτα SD και περιέχει όλα τα παραπάνω λειτουργικά. Στο πρώτο άνοιγμα του Raspberry Pi αποφασίζει ο χρήστης ποιο από τα λειτουργικά θέλει να εγκαταστήσει και τα υπόλοιπα διαγράφονται αυτόματα από την κάρτα. (στο παράρτημα υπάρχει οδηγός εγκατάστασης).

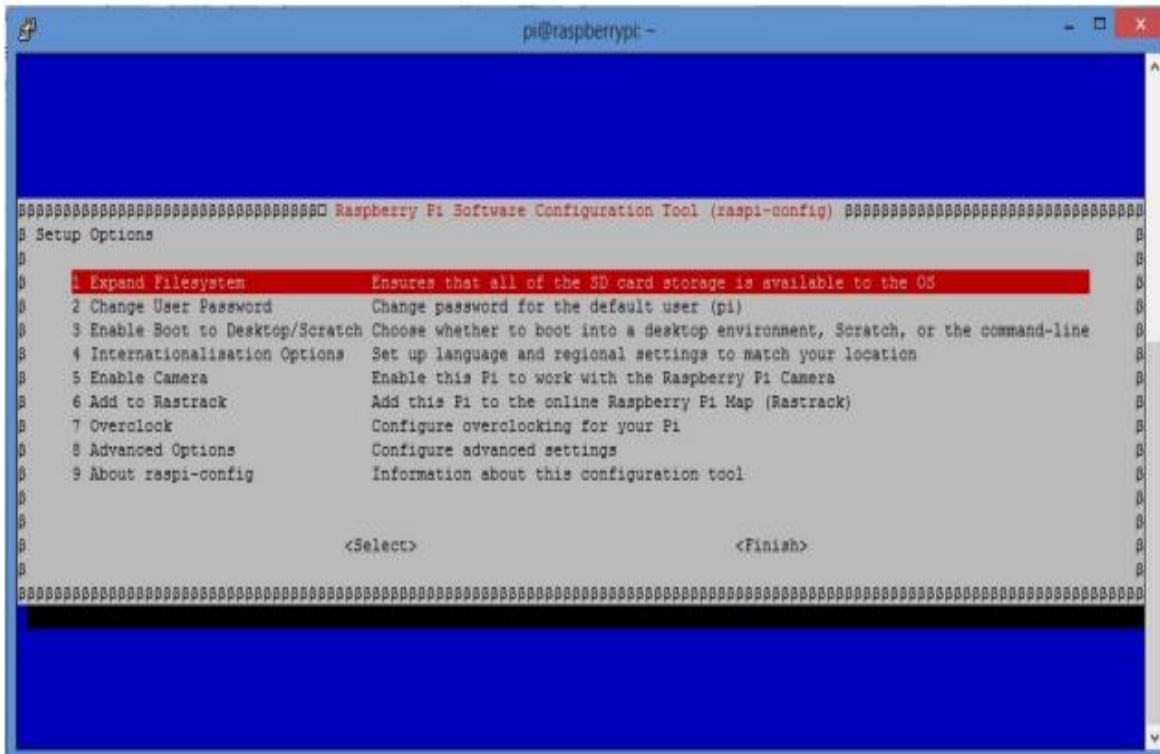
### 3.2.2 Στήσιμο Μηχανημάτων

Το λειτουργικό που επιλέχθηκε είναι το Noobs καθώς είναι η προτεινόμενη εγκατάσταση για αρχάριους. Για να εγκαταστήσουμε το λειτουργικό Noobs στο Raspberry Pi θα πρέπει να ακολουθήσουμε τα παρακάτω βήματα:

Βάζουμε την κάρτα σε έναν υπολογιστή που στην συγκεκριμένη περίπτωση τρέχει Windows 7 και με το SD Formatter κάνουμε format στην κάρτα έχοντας την επιλογή size adjustment στο ON έτσι να γίνει προσαρμογή στο μέγεθος της κάρτας. Έπειτα, με το Win32DiskImager αντιγράφουμε το αρχείο image στην κάρτα. Οι δύο παραπάνω διαδικασίες μπορούν να γίνουν και με άλλα αντίστοιχα προγράμματα που χρησιμοποιούνται για την διαμόρφωση καρτών SD.

Στη συνέχεια, βάζουμε την κάρτα SD στο Raspberry Pi και αφού το έχουμε συνδέσει στο τοπικό μας δίκτυο το ανάβουμε. Οι αρχικές ρυθμίσεις του Raspberry Pi είναι η απόκτηση IP να γίνεται μέσω dhcp client και η δυνατότητα σύνδεσης ssh να είναι ενεργοποιημένη. Έτσι, βρίσκουμε την IP διεύθυνση του Raspberry Pi, από τις συνδεδεμένες συσκευές στο Router, που από εδώ και στο εξής θα την αναφέρουμε ως IP address . Συνδεόμαστε στην πλακέτα απομακρυσμένα με ssh με ένα πρόγραμμα όπως το Putty. Αν όλα έχουν πάει καλά η σύνδεση είναι επιτυχής και εμφανίζεται στον υπολογιστή μας ένα παράθυρο με την λέξη "login as:". Το προκαθορισμένο username και password είναι pi και raspberry αντίστοιχα.

Η πρώτη ρύθμιση που θα πρέπει να κάνουμε στο Raspberry Pi είναι να επιτρέπεται η χρήση κάμερας (enable camera). Για να εισέλθουμε στις ρυθμίσεις του Raspberry Pi εκτελούμε την εντολή `sudo raspi-config` (Επισημαίνεται ότι για τον χρήστη pi δεν χρειάζεται κωδικός για να εκτελεστεί μία εντολή με sudo). Με την εκτέλεση αυτής της εντολής βλέπουμε τις παρακάτω επιλογές:



Εικόνα 24 – Επιλογές Ρύθμισης Raspberry pi

Επιλέγουμε το 5 Enable Camera και πατάμε Enter. Μετά το τέλος της διαδικασίας αυτής μπορούμε να έχουμε τις λειτουργίες της κάμερας.

Στη συνέχεια εκτελούμε την εντολή `sudo apt-get update && sudo apt-get upgrade` (απαντάμε με YES στο ερώτημα αν θέλουμε να συνεχίσουμε) έτσι ώστε να γίνουν όλες οι τελευταίες αναβαθμίσεις που είναι διαθέσιμες για το Raspberry Pi καθώς επίσης και την εντολή `sudo apt-get clean` για να διαγραφούν τα προσωρινά αρχεία της εγκατάστασης.

Στη συνέχεια εγκαθιστούμε τις απαραίτητες βιβλιοθήκες για την εφαρμογή μας. Από την γραμμή εντολών εκτελούμε τις παρακάτω εντολές:

```
sudo apt-get install python-dev
```

```
sudo apt-get install python-rpi.gpio
```

```
sudo apt-get install python-pip
```

```
sudo pip install pil
```

Καθόλη τη διάρκεια της εργασίας πλην της φάσης της εγκατάστασης συνδεόμουν στο Raspberry μέσω του πρωτοκόλλου ssh.

Περισσότερες και πιο αναλυτικές πληροφορίες σχετικά με την εγκατάσταση υπάρχουν στο παράρτημα.

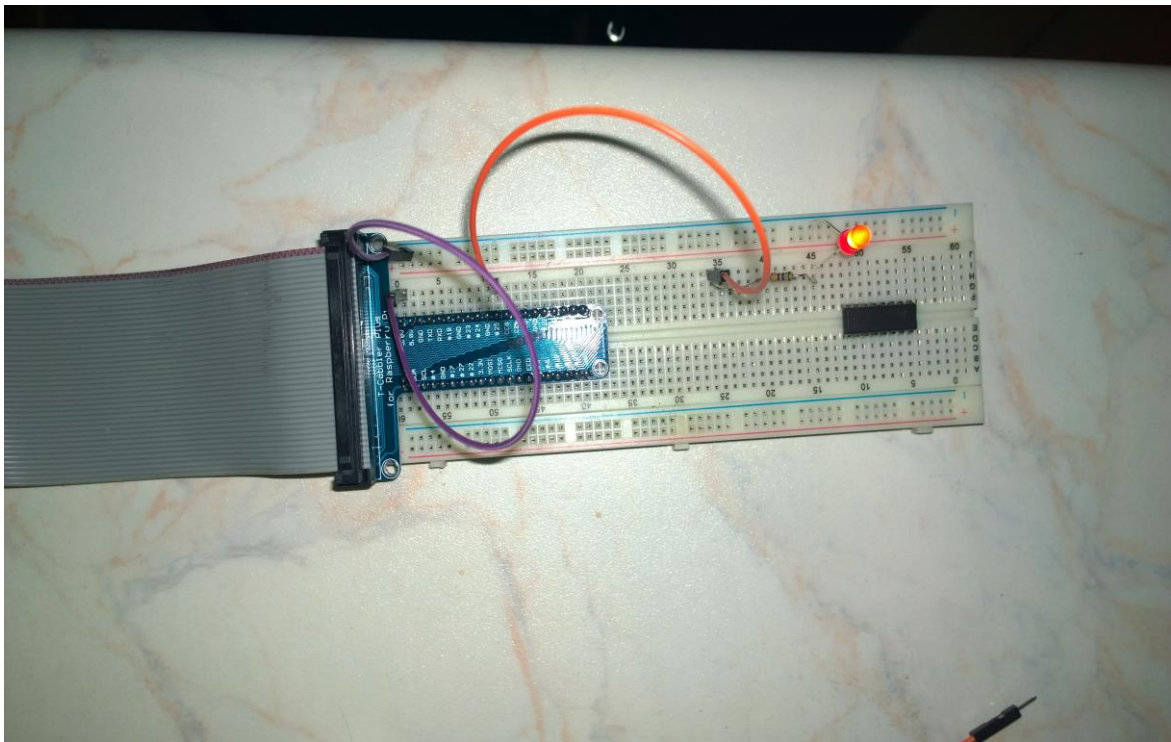


## Κεφάλαιο 4

### 4.1 Υλοποίηση Κατασκευής

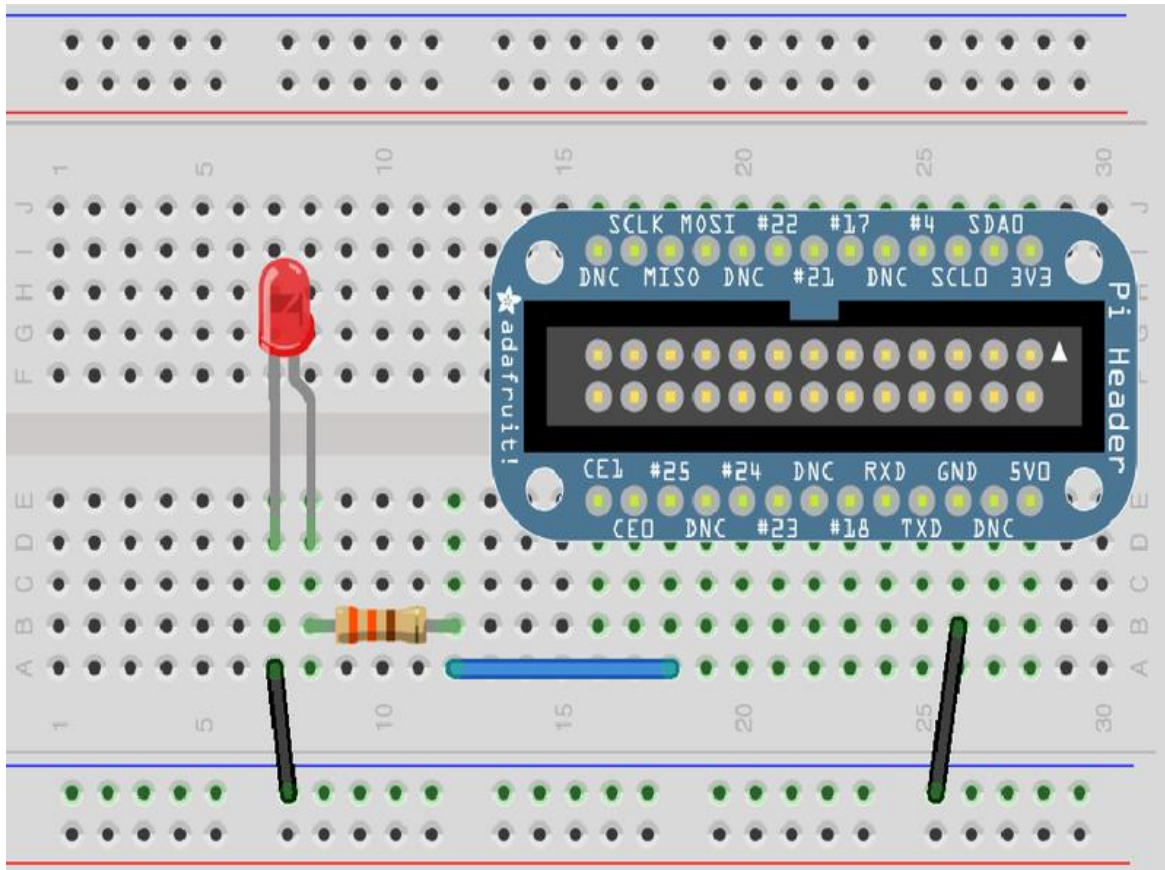
#### 4.1.1 Στήσιμο πρώτου κυκλώματος

Στην παρακάτω εικόνα εμφανίζεται το πρώτο κύκλωμα που δημιουργήθηκε για την εξοικείωση με το περιβάλλον του Raspberry Pi. Με την εκτέλεση του κώδικα ανάβει το led το οποίο βρίσκεται πάνω στο breadboard. Η εκτέλεση του κώδικα πραγματοποιείται αν γράψουμε στο terminal του raspberry την εντολή `sudo python led.py`. Ο κώδικας βρίσκεται στο παράρτημα(led.py).



Εικόνα 25 - Κύκλωμα Raspberry pi για ενεργοποίηση Led

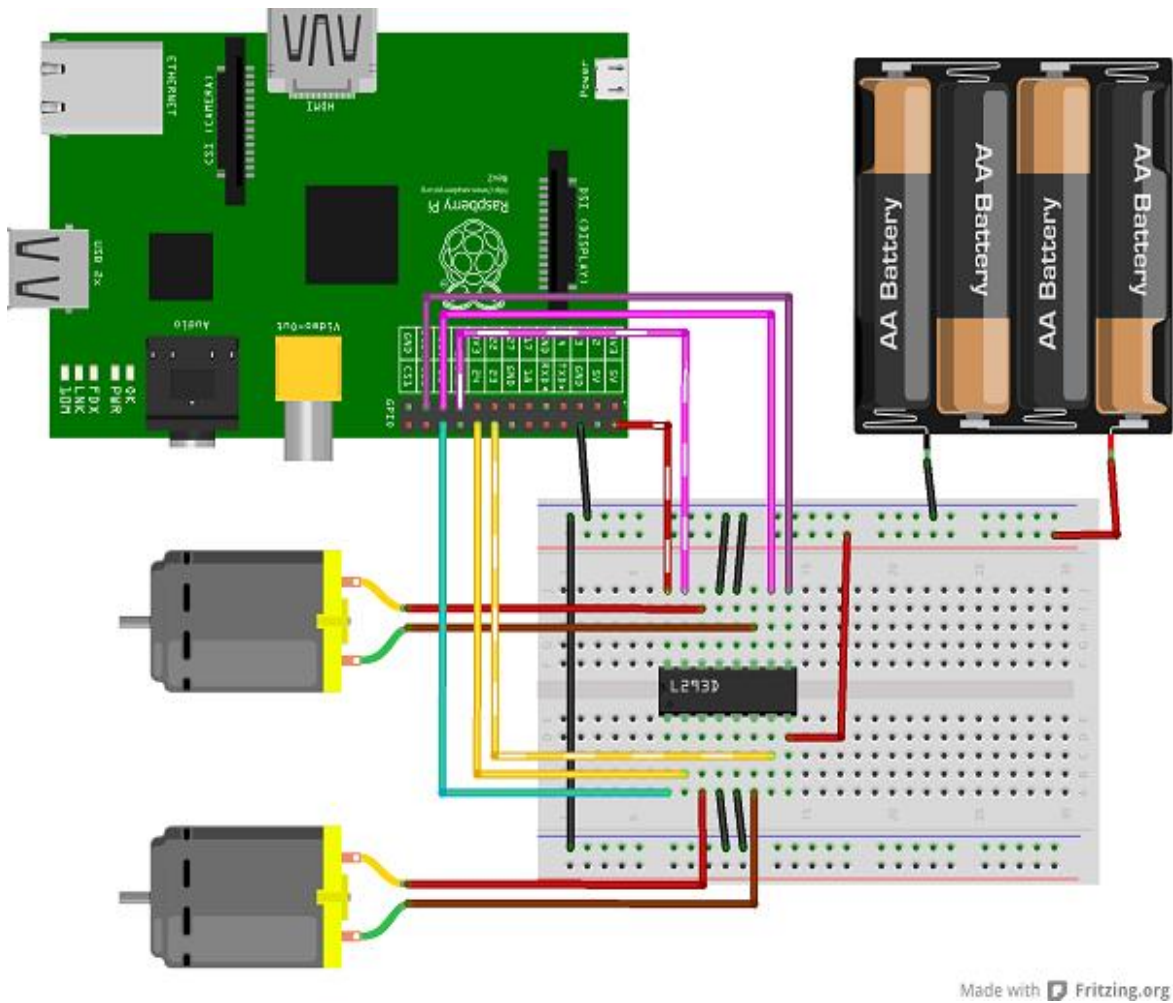




Εικόνα 26 - Κύκλωμα ενεργοποίησης Led

Κατά την εκτέλεση του κώδικα led.py (sudo python led.py) η τιμή του GPIO PIN 25 (σε BCM mode) αλλάζει από 0 σε 1. Μένει 1 για 10 δευτερόλεπτα και στη συνέχεια αλλάζει πάλι σε 0 και σβήνει.

#### 4.1.2 Στήσιμο Δεύτερου Κυκλώματος



Εικόνα 27 - Κύκλωμα για τον έλεγχο δύο ηλεκτρικών κινητήρων



Εικόνα 28 - Κύκλωμα ελέγχου ηλεκτρικών κινητήρων

Στην παραπάνω εικόνα εμφανίζεται η συνδεσμολογία για έλεγχο δύο ηλεκτρικών κινητήρων από το raspberry pi (κώδικας dcmotor.py στο παράρτημα). Στον κώδικα πραγματοποιήθηκαν έλεγχοι για τον καθορισμό των παρακάτω κινήσεων.

### Στροφή Δεξιά

Για την στροφή δεξιά πρέπει οι τιμές των GPIO PINS (σε BCM Mode) να είναι όπως παρακάτω:

```
def right():
    print "right"
    GPIO.output(19, GPIO.LOW)
    GPIO.output(21, GPIO.HIGH)
    GPIO.output(23, GPIO.HIGH)
```

### Στροφή Αριστερά

Για την στροφή αριστερά πρέπει οι τιμές των GPIO PINS (σε BCM Mode) να είναι όπως παρακάτω:

```
def left():
    print "turn left"
    GPIO.output(19, GPIO.HIGH)
    GPIO.output(21, GPIO.LOW)
    GPIO.output(23, GPIO.HIGH)
```

### Κίνηση Ευθεία Μπροστά

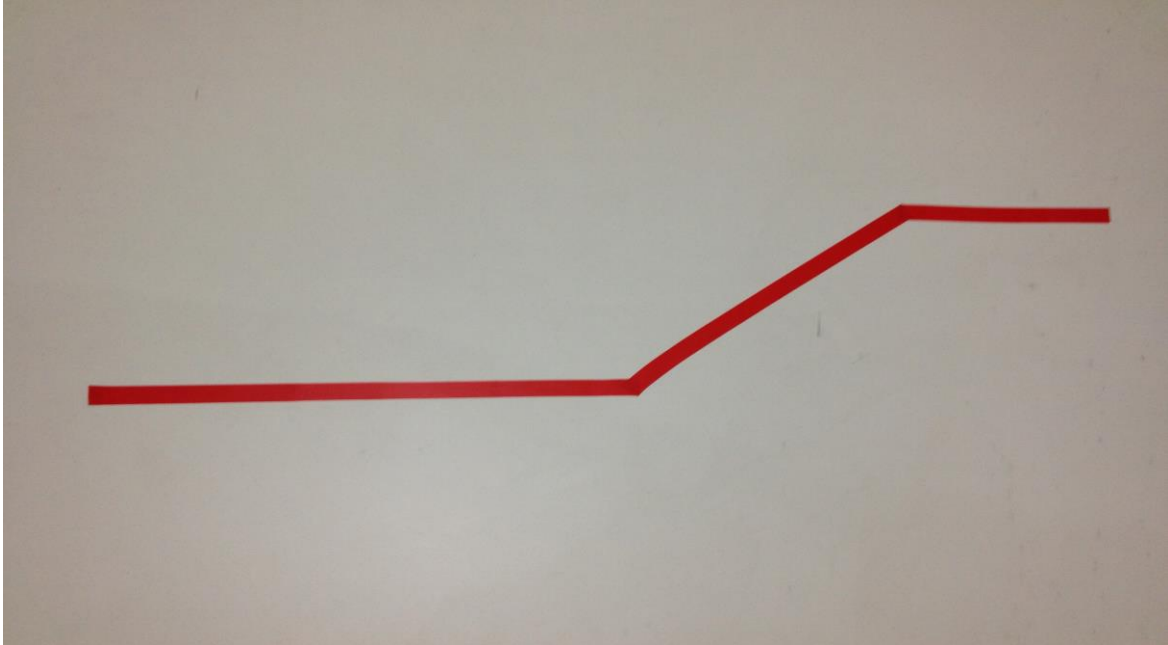
Για την κίνηση ευθεία μπροστά πρέπει οι τιμές των GPIO PINS (σε BCM Mode) να είναι όπως παρακάτω:

```
def go():
    print "go"
    GPIO.output(22, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(16, GPIO.HIGH)
```

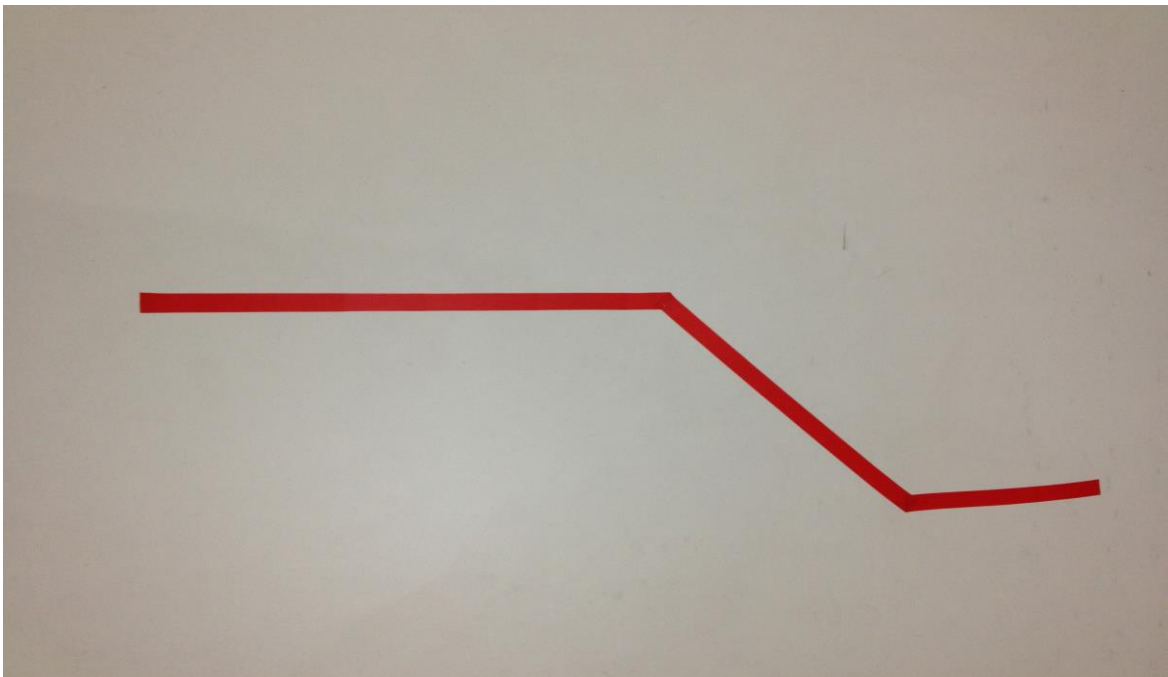
Εικόνες και βίντεο υπάρχουν διαθέσιμα στο cd της πτυχιακής.

#### 4.1.3 Στήσιμο Πίστας

Για την δημιουργία πίστας δοκιμών χρησιμοποιήθηκε ένα λευκό πάτωμα και μία κόκκινη λωρίδα στη μέση. Παρακάτω εμφανίζονται δύο σενάρια που δοκιμάστηκαν, μία αριστερή και δεξιά στροφή και μία δεξιά και αριστερή στροφή.

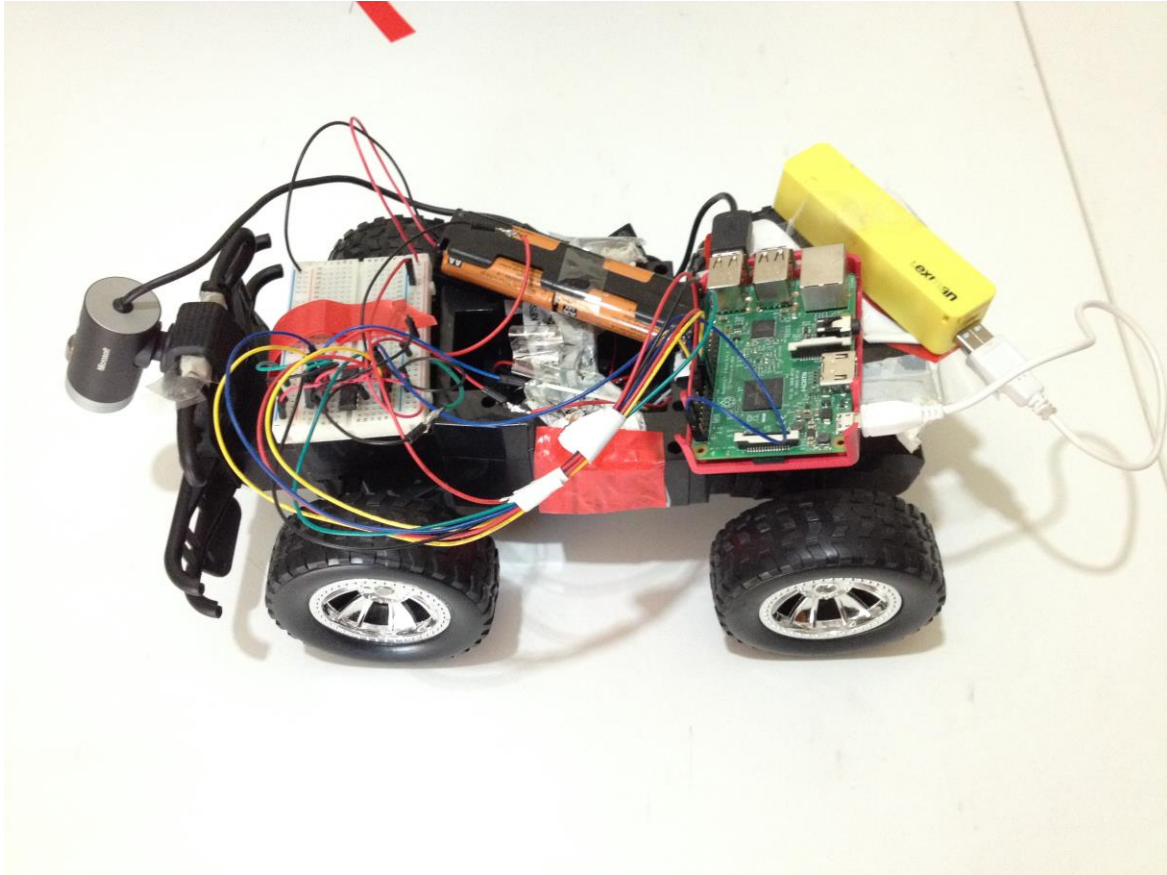


Εικόνα 29 – Αριστερή/Δεξιά στροφή



Εικόνα 30 – Δεξιά/Αριστερή στροφή

#### 4.1.4 Στήσιμο Οχήματος



Εικόνα 31 – Όχημα με Raspberry Pi & Usb Κάμερα

Σε αυτήν την ενότητα θα περιγραφούν όλα τα βήματα τα οποία είναι απαραίτητα για κατασκευή του παραπάνω οχήματος.

- Προμήθεια όλων των υλικών(Raspberry, Camera, Breadboard κ.ά)
- Εργαλεία ηλεκτρονικής (για παράδειγμα πένσα, κολλητήρι κ.α.)
- Ξεβίδωμα του επάνω μέρους του οχήματος
- Τοποθέτηση και σφίξιμο στα καλώδια των ηλεκτρικών κινητήρων συνδετικών για breadboard
- Στήσιμο του κυκλώματος της εικόνας 27
- Αφού συνδέσουμε το raspberry pi με το breadboard και τους dc motors τότε τροφοδοτούμε το raspberry(Εκκίνηση raspberry)
- Σύνδεση του raspberry στο router μέσω Wifi

- Εκτέλεση του προγράμματος `python dcmotor.py` με την εντολή `sudo python dcmotor.py` (Έλεγχος εάν οι κινητήρες λειτουργούν σύμφωνα με τα μηνύματα που τυπώνονται στην οθόνη)
  - Απενεργοποίηση του Raspberry
  - Σύνδεση της Usb Camera με το Raspberry Pi
  - Εκκίνηση του Raspberry Pi
  - Τοποθέτηση μίας σελίδας A4, με μία κόκκινη γραμμή πάνω μπροστά από την κάμερα
  - Εκτέλεση του προγράμματος `linedetect.py` (`sudo python linedetect.py`), για να δούμε αν αναγνωρίζει την γραμμή και αν κινείται σύμφωνα με την γραμμή.
  - Επανάλαβε την εκτέλεση με διαφορετικές σελίδες, διαφορετικές γραμμές (π.χ. διαφορετική κατεύθυνση, φωτισμό, πάχος γραμμής), διαφορετική ανάλυση (resolution)
    - Στη συνέχεια και αφού τα αποτελέσματα είναι ικανοποιητικά, συναρμολόγησε το όχημα
    - Τοποθέτηση του power bank στο όχημα και τροφοδοσία του Raspberry pi(Εικόνα στο παράρτημα)
    - Τροφοδοσία του κυκλώματος του breadboard μέσω του αυτοσχέδιου power bank(Εικόνα στο παράρτημα)
    - Δημιουργία μίας μεγαλύτερης πίστας
    - Τοποθέτηση του οχήματος στην πίστα
    - Σύνδεση στο raspberry μέσω ssh (putty)
    - Εκτέλεση του προγράμματος `linedetect.py`

Αφού εκτελέσουμε τα παραπάνω βήματα τότε το όχημα είναι έτοιμο τρέξει σε οποιαδήποτε πίστα η οποία έχει μια κόκκινη γραμμή στην μέση (όπως στις εικόνες 29 και 30).Όταν εκτελέσουμε το πρόγραμμα `linedetect.py` τότε θα εμφανιστεί στην οθόνη ένα παράθυρο που θα δείχνει τι τραβάει η κάμερα (δηλαδή την γραμμή αν είναι στην πίστα), καθώς και ένας κύκλος στο μέσο της γραμμής. Το βίντεο που θα εμφανιστεί στο παράθυρο σε λειτουργικό Linux λειτουργεί σωστά, ενώ σε Windows υπάρχει μια μικρή καθυστέρηση. Στο τερματικό θα εμφανίζονται οι συντεταγμένες της γραμμής  $x,y$  καθώς και αν θα στρίβει το όχημα ή θα πηγαίνει ευθεία (`move left, move right ,go`).

## Κεφάλαιο 5

### 5.1 Αποτελέσματα Εργασίας

Τα αποτελέσματα ήταν ικανοποιητικά στις πίστες των εικόνων 29 και 30 και το όχημα ακολουθούσε και έστριβε σύμφωνα με την γραμμή αλλά υπήρχαν και περιπτώσεις που σε πιο περίπλοκες διαδρομές ή σε διαδρομές με πολλαπλές γραμμές οι αποφάσεις του αλγόριθμου να μην είναι πάντα σωστές.

Το όχημα δεν έχει τη δυνατότητα να αναγνωρίσει εμπόδια. Επιπλέον σε σταυροδρόμια επιλέγει να πάει ευθεία (ανάλογα με τη φορά της κάμερας).

### 5.2 Δυσκολίες

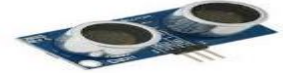
Κατά την αποπεράτωση της πτυχιακής συνάντησα τις παρακάτω δυσκολίες:

- Λόγω των “capital controls”, συνάντησα ελλείψεις σε υλικά τα οποία ήταν απαραίτητα για την υλοποίηση της εργασίας.
- Τα καταστήματα τα οποία πουλάνε είδη ηλεκτρονικής ρομποτικής είναι περιορισμένα.
- Κατά την υλοποίηση λόγω ότι έδινα τιμές στα GPIO pins, raspberry παρουσίαζε δυσλειτουργίες (π.χ. έχανε το δίκτυο ή έκανε reboot σε safe mode).
- Μετά το πέρας της εκτέλεσης των προγραμμάτων οι τιμές των GPIO PINS δεν επανέρχονταν στην αρχική κατάσταση
- Όταν δοκίμασα την βιβλιοθήκη openCV και live streaming στην κάμερα, το raspberry δεν ανταποκρινόταν μετά από συνεχείς δοκιμές
- Κατά την διάρκεια των πολλών δοκιμών που έκανα κάποια εξαρτήματα χάλασαν (όπως το raspberry pi, το breadboard και οι κινητήρες) με αποτέλεσμα να αγοράσω καινούργια



### 5.3 Βελτιώσεις

Για την βελτίωση της ακρίβειας κίνησης του οχήματος θα μπορούσαν να γίνουν τα παρακάτω:



- Προσθήκη στο όχημα ενός distance sensor δηλαδή ενός αισθητήρα απόστασης
- Χρήση motor controller (2A Dual Motor Controller)
- Συγγραφή κώδικα avoid obstacle detection δηλαδή ενός κώδικα ανίχνευσης και αποφυγής εμποδίων
- Χρήση καλύτερου οχήματος
- Χρήση πιο αποδοτικών κινητήρων (dc motor)

## ΠΑΡΑΡΤΗΜΑ

### Ειδικό Λεξιλόγιο (Glossary) και Συντομογραφίες

Open source - ανοιχτού κώδικα

Scripting language - περιγραφική γλώσσα

Rapid development - γρήγορη ανάπτυξη

Linux - λειτουργικό σύστημα το οποίο βασίζεται στο Unix

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

BCM - Broadcom SOC channel

Username - όνομα χρήστη

Password - κωδικός πρόσβασης

OpenCV - βιβλιοθήκη για επεξεργασία εικόνας

Live streaming - ζωντανή μετάδοση

Distance sensor - ανιχνευτής απόστασης

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

Power bank - αποθήκη ενέργειας

Breadboard - ολοκληρωμένο

Board - πλακέτα

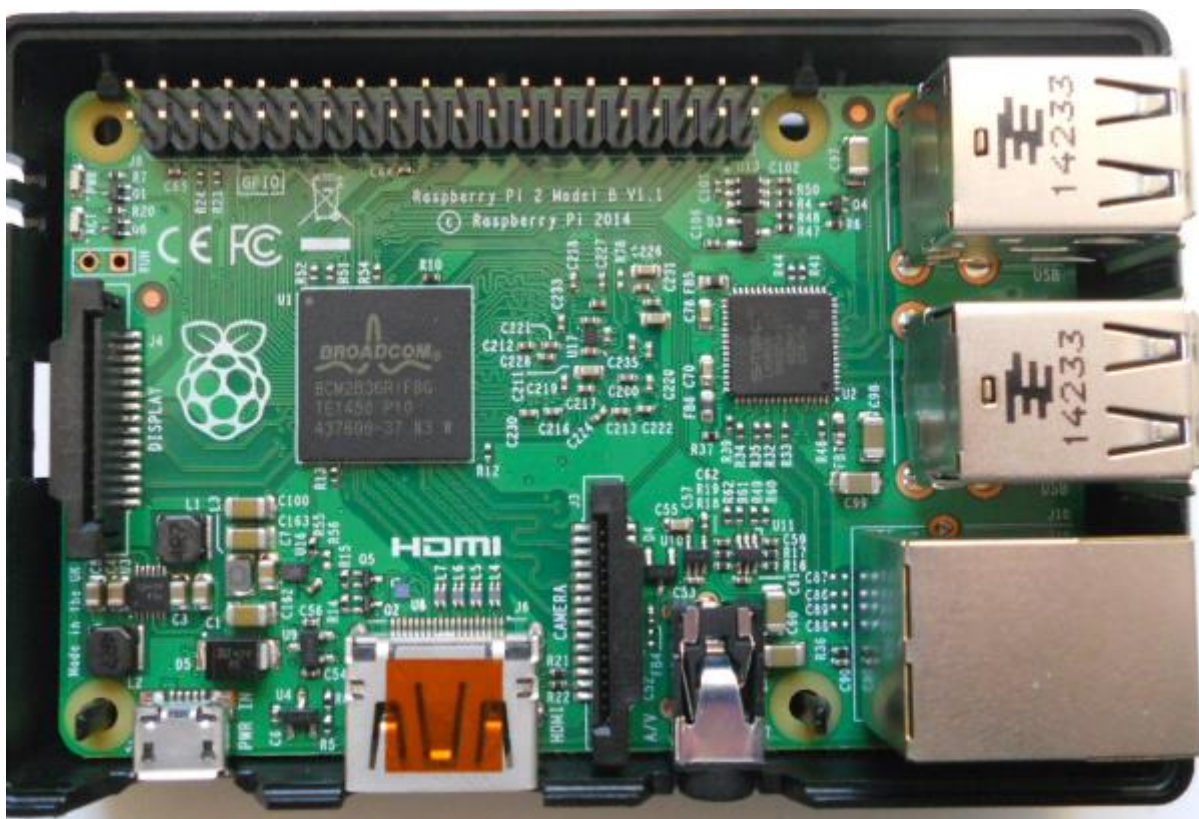
## Εγκατάσταση Λειτουργικού Noobs

Η εγκατάσταση λειτουργικού στο raspberry είναι πολύ απλή διαδικασία. Μεταφορτώνουμε τα αρχεία του Noobs από το διαδικτυακό τόπο <https://www.raspberrypi.org/downloads/noobs/>. Αναλυτικά για την εγκατάσταση του Noobs ήταν απαραίτητα τα παρακάτω υλικά και βήματα.

Τα υλικά που είναι απαραίτητα για την εγκατάσταση των Noobs για το Raspberry Pi τα έχουμε αναφέρει στο Κεφάλαιο 3 και θα τα δούμε σε παρακάτω εικόνες.

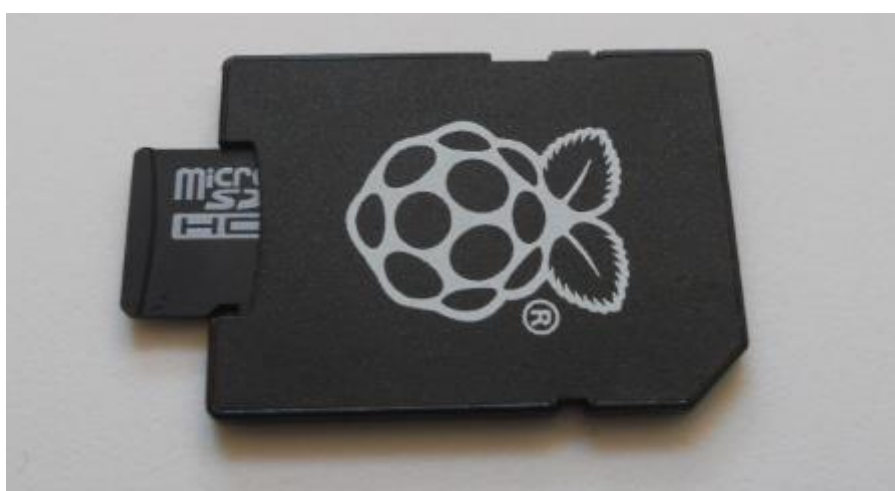


Εικόνα 32 - Υλικά Εγκατάστασης Λειτουργικού



Εικόνα 33 - Raspberry Pi

Έχουμε επίσης μια κάρτα Micro SD, η οποία θα πρέπει να έχει μέγεθος τουλάχιστον 4GB και να είναι τουλάχιστον Class 4.



Εικόνα 34 - Micro SD

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

Η τροφοδοσία για το Raspberry Pi είναι Micro USB, ίδια με των κινητών. Προσοχή, όμως, αν θέλουμε να συνδέσουμε στο Raspberry Pi κάποιον εξωτερικό σκληρό με USB, ή γενικά οποιαδήποτε συσκευή έχει αυξημένες ανάγκες σε ρεύμα, θα πρέπει η τροφοδοσία να μπορεί να δώσει επαρκή Ampere. Το Raspberry Pi 2 και το Raspberry Pi 3 μπορούν να δώσουν maximum μέχρι 2 Ampere με την κατάλληλη τροφοδοσία.



Εικόνα 35 - Απαραίτητη τάση για Raspberry Pi 3

Ένα τελευταίο απαραίτητο εξάρτημα, αν δεν έχουμε laptop με Card Reader, είναι ένας Reader για τον σταθερό μας υπολογιστή.



**Εικόνα 36 - SD Card Reader**

Αν και δεν εικονίζονται, θα χρειαστούμε καλώδιο HDMI-HDMI ή HDMI-DVI, καλώδιο Ethernet, καθώς και USB πληκτρολόγιο και ποντίκι. Μπορούμε επίσης να συνδεθούμε και ασύρματα μέσω Wifi, αν σκοπεύουμε να τοποθετήσουμε το Raspberry Pi μακριά από το router (δρομολογητή).

## Εγκατάσταση NOOBS στην κάρτα SD

Το NOOBS, από τα αρχικά της φράσης New Out Of Box Software - "καινούριο software, του κουτιού", σε ελεύθερη μετάφραση - είναι ένα σύστημα που διευκολύνει σημαντικά την αρχική εγκατάσταση Raspberry Pi για τους αρχάριους. Το πρώτο που χρειάζεται είναι να κάνουμε είναι να συνδέσουμε την κάρτα Micro SD με τον ανάπτορα στον υπολογιστή μας, και να τη διαμορφώσουμε κατάλληλα. Το ίδιο το NOOBS συνιστά να χρησιμοποιήσουμε την επίσημη εφαρμογή για Format σε SD, την οποία θα βρούμε στη διεύθυνση:

- [https://www.sdcard.org/downloads/formatter\\_4/eula\\_windows/](https://www.sdcard.org/downloads/formatter_4/eula_windows/)

### Article 7 Export Control

Licensee agrees not to export or re-export to any country the Software in any form without the appropriate export licenses under regulations of the country where Licensee is located, if necessary.

### Article 8 Termination of License

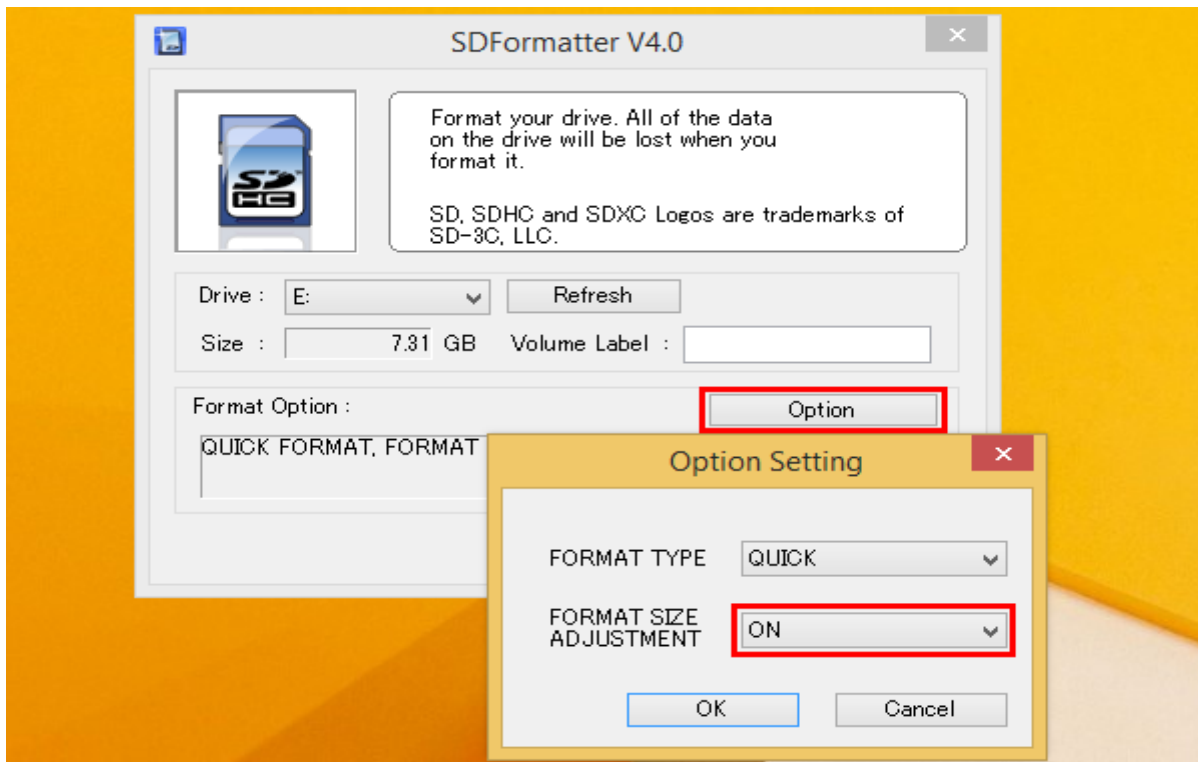
The rights granted to Licensee hereunder will be automatically terminated if Licensee contravenes any of the terms and conditions of this Agreement. In the event of termination, Licensee must destroy the Software and related documentation together with all the copies thereof at Licensee's own expense.



Εικόνα 37 - Κατέβασμα SD Formatter

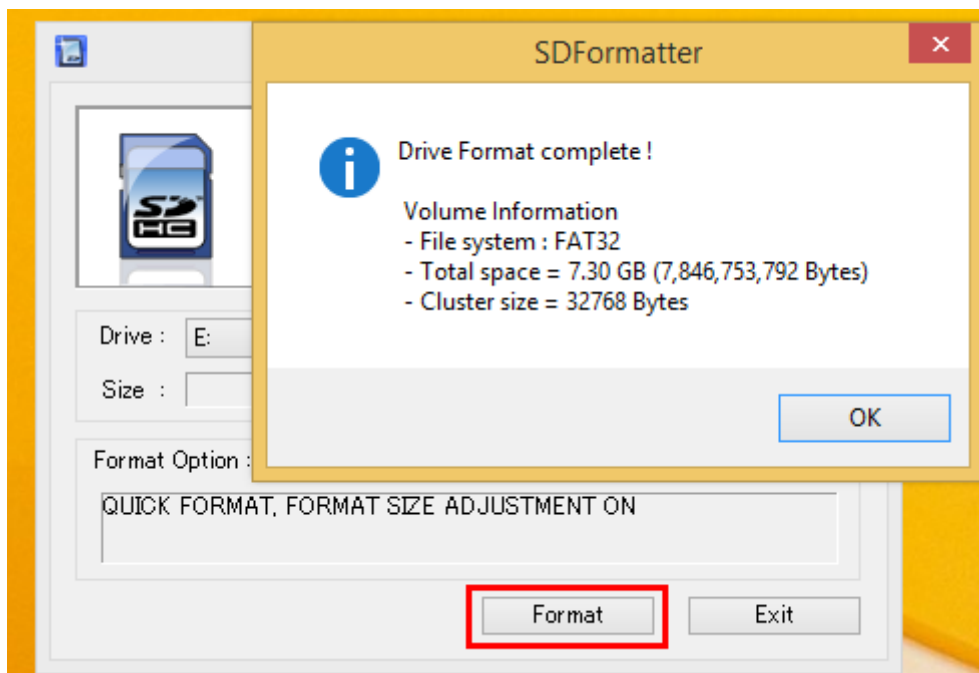
Κατεβάζουμε την εφαρμογή κάνοντας κλικ στο Accept στο κάτω μέρος της σελίδας. Αποσυμπιέζουμε το αρχείο .zip, το εγκαθιστούμε και τρέχουμε την εφαρμογή - η εγκατάσταση δεν έχει καμία δυσκολία - και επιλέγουμε το "Option", στο οποίο αλλάζουμε το Format Size Adjustment σε "ON".

Παρακάτω βλέπουμε και τις εικόνες με τις ρυθμίσεις που αλλάζουμε στο SD Formatter.



**Εικόνα 38 – Ρυθμίσεις SD Formatter**

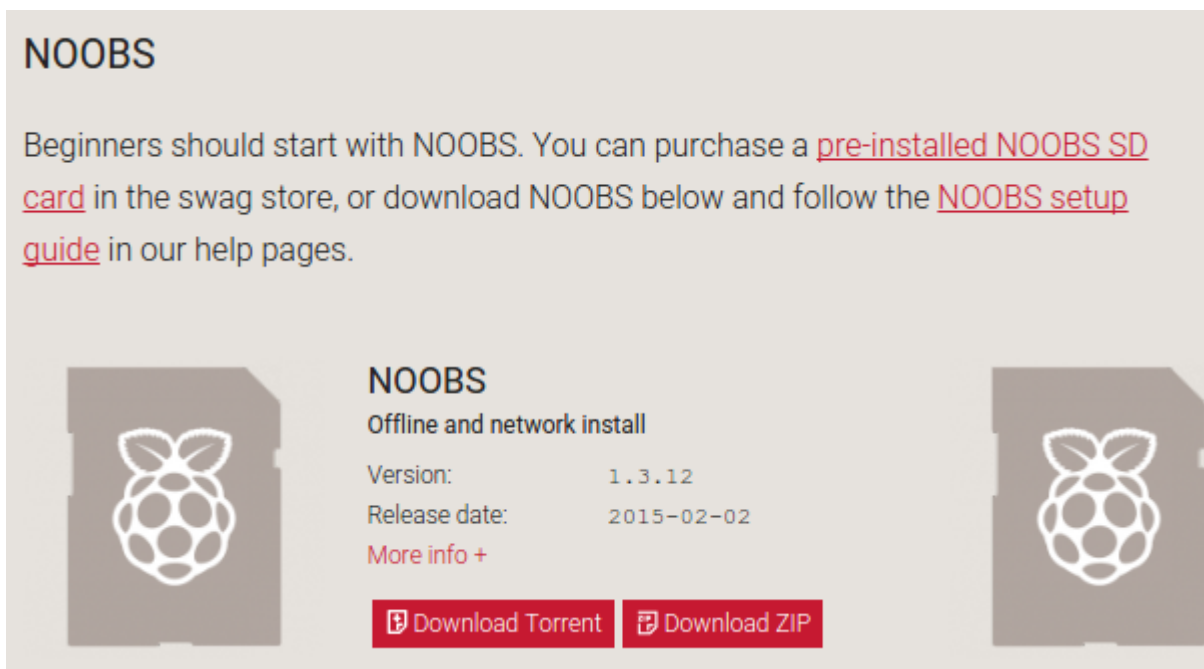
Κάνουμε κλικ στο Format, και σε λίγα δευτερόλεπτα έχει ολοκληρωθεί.



**Εικόνα 39 – Ολοκλήρωση SD Formatter**

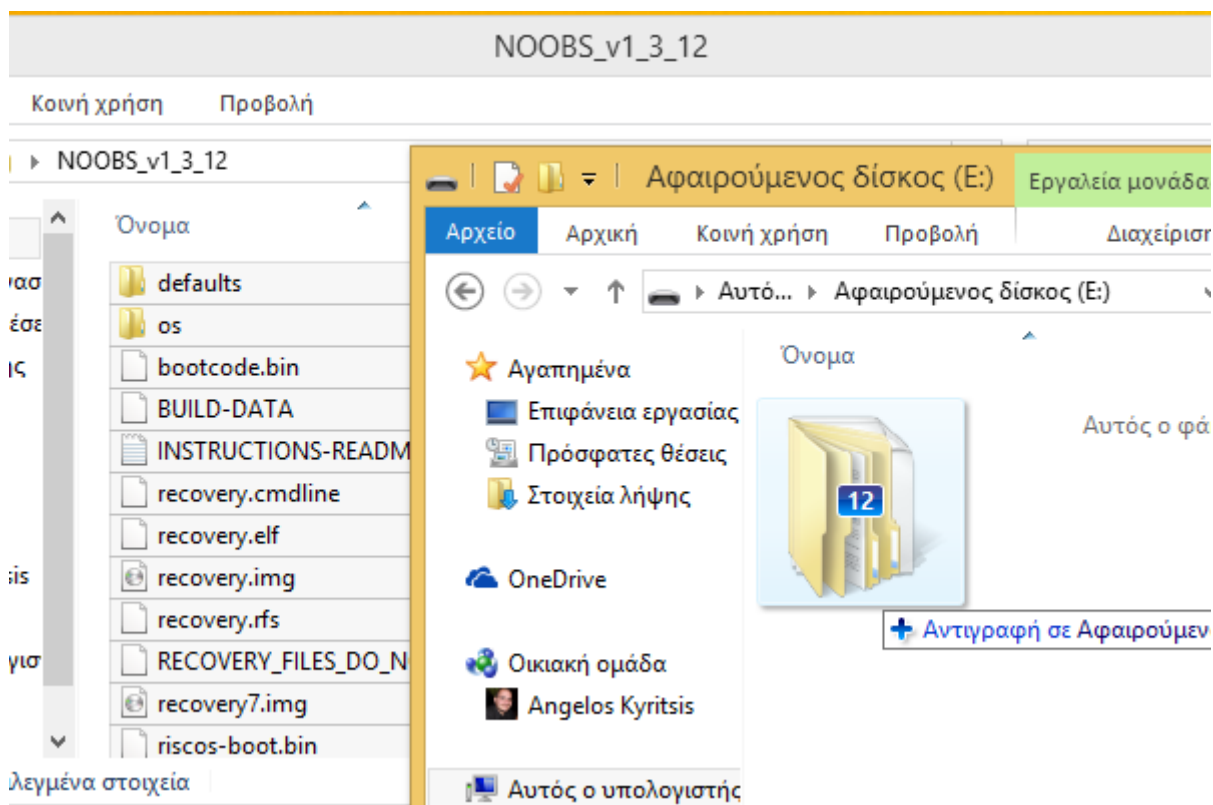


Με την κάρτα SD μας διαμορφωμένη, μπαίνουμε στη διεύθυνση <http://www.raspberrypi.org/downloads/> και κατεβάζουμε την τελευταία έκδοση του NOOBS, είτε σε .zip είτε μέσω Torrent. Έχει μέγεθος λίγο παραπάνω από 730MB. Γενικά το κατέβασμα με Torrent είναι σημαντικά πιο γρήγορο από το κατέβασμα μέσω της ιστοσελίδας.



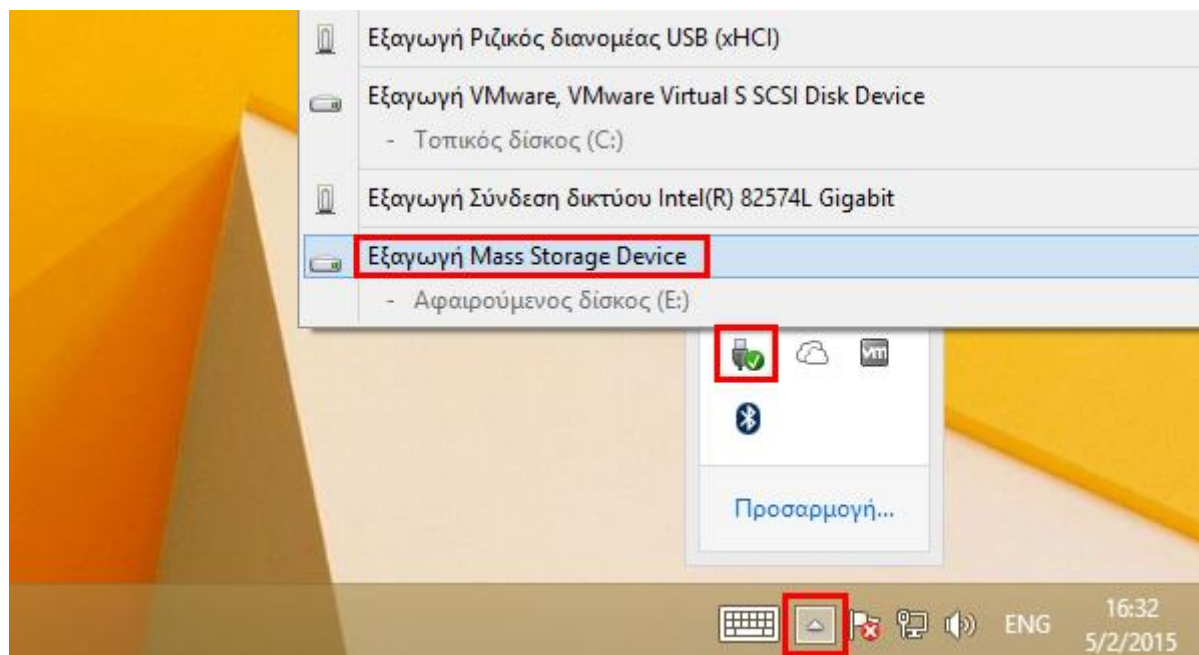
**Εικόνα 40 - Noobs Official Page**

Αφού κατεβάσουμε το συμπιεσμένο φάκελο .zip, κάνουμε αποσυμπίεση σε όλα τα περιεχόμενά του. Ανοίγουμε το φάκελο στον οποίο έγιναν αποσυμπίεση, και τα αντιγράφουμε στην κάρτα SD, όπως βλέπουμε στην παρακάτω εικόνα.



Εικόνα 41 – Αντιγραφή αρχείων στην SD card

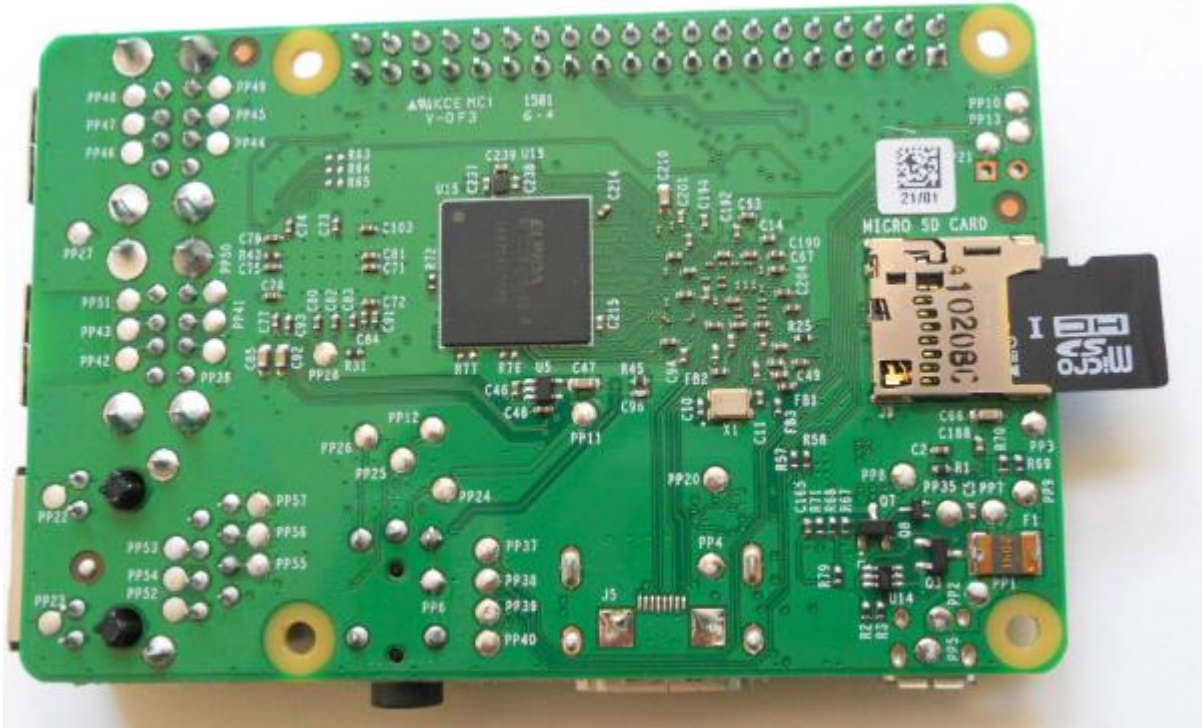
Αφού ολοκληρωθεί η αντιγραφή, κάνουμε εξαγωγή της κάρτας SD.



Εικόνα 42 - Εξαγωγή SD card

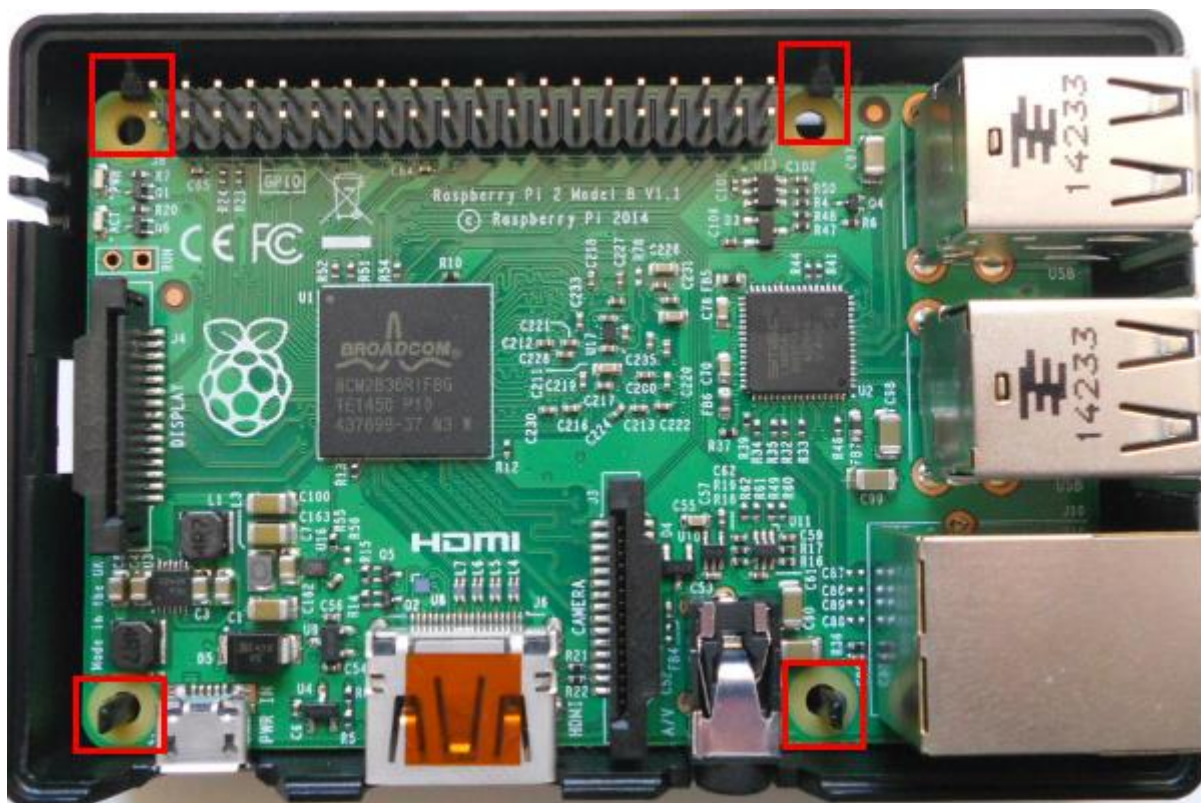
## Εγκατάσταση Raspberry Pi

Κρατάμε το Raspberry Pi και το γυρίζουμε ανάποδα για να τοποθετήσουμε την Micro SD. Υπάρχει ένας μόνο τρόπος για να μπει, και μπαίνοντας θα "κλειδώσει".



Εικόνα 43 - Είσοδος SD στο Raspberry Pi

Για να αφαιρέσουμε την SD, την ξαναπατάμε προς τα μέσα, για να ξεκλειδώσει. Αν μαζί με το Raspberry Pi αγοράσαμε και θήκη, καλό είναι να το τοποθετήσουμε πριν βάλουμε την κάρτα, γιατί έστω και το λίγο που προεξέχει μπορεί να μας δυσκολέψει.



**Εικόνα 44 – Θήκη Raspberry Pi**

Στη συγκεκριμένη θήκη πρώτα χρειάστηκε να βάλουμε υπό γωνία το Raspberry Pi ώστε να μπει στις επάνω υποδοχές, και στη συνέχεια το κατεβάσαμε για να μπουν τα αντίστοιχα στελέχη στις κατάλληλες οπές. Με το Raspberry Pi στη θήκη του, βάζουμε την Micro SD από το σχετικό άνοιγμα.



**Εικόνα 45 – Είσοδος SD Card στη θήκη**

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

Για τα άλλα καλώδια, το μόνο σημαντικό είναι να βάλουμε την τροφοδοσία τελευταία. Βλέπετε, το Raspberry Pi δεν έχει διακόπτη On/Off. Μόλις το βάλουμε στην πρίζα, θα ξεκινήσει κατευθείαν.

Βάζουμε πρώτα το καλώδιο HDMI...



**Εικόνα 46 - HDMI Καλώδιο**

...και στη συνέχεια το Ethernet, το πληκτρολόγιο και το ποντίκι.



**Εικόνα 47 - Πληκτρολόγιο και Ποντίκι**

Αν έχουμε συνδέσει το Raspberry Pi στην τηλεόραση, επιλέγουμε σαν είσοδο το HDMI, και συνδέουμε το Raspberry Pi στην πρίζα.

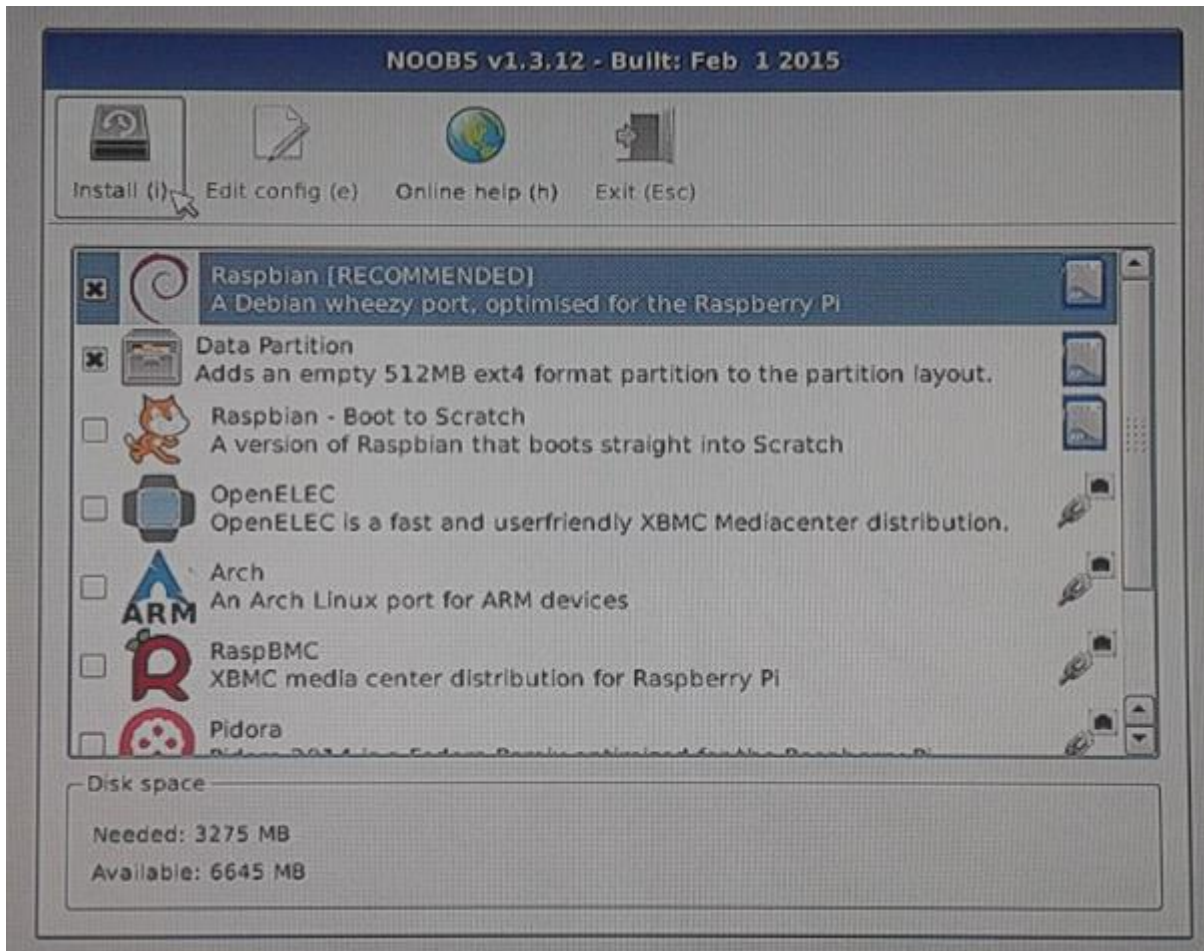
### **Πρώτη εκκίνηση του Raspberry Pi**

Ξεκινώντας, θα μας δείξει μια οθόνη με διάφορα χρώματα, γνωστή και σαν rainbow screen.



**Εικόνα 48 – Εκκίνηση Raspberry Pi**

(Οι φωτογραφίες σε αυτό το σημείο δεν έχουν και την καλύτερη ποιότητα, καθώς φωτογράφιζα την οθόνη). Σύντομα θα φορτώσει το NOOBS, το οποίο θα μας εμφανίσει μια λίστα με τα λειτουργικά συστήματα.



**Εικόνα 49 – Λειτουργικά Εγκατάστασης**

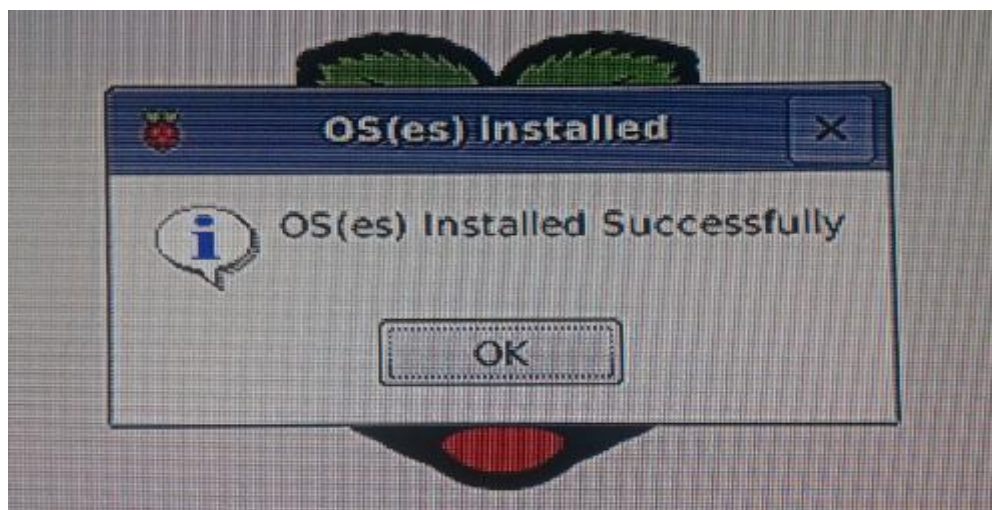
Στο παράδειγμα θα εγκαταστήσουμε το Raspbian, μια παραλλαγή του Debian για το Raspberry Pi, η οποία μπορεί να εγκατασταθεί και χωρίς σύνδεση στο Internet. Οι περισσότερες από τις υπόλοιπες επιλογές (OpenELEC, Arch, RaspBMC κλπ) χρειάζονται σύνδεση. Έχοντας τσεκάρει το Raspbian και κάνοντας κλικ στο Install, το σύστημα μας προειδοποιεί πως θα διαγραφεί όλο το περιεχόμενο της SD.

Επιλέγοντας "Yes", ξεκινάει η εγκατάσταση. Ανάλογα με την ταχύτητα της κάρτας SD που έχουμε βάλει, θα πάρει αρκετή ώρα. Η συγκεκριμένη SD, που ήταν Class 4, έγραφε με τον ρυθμό των 1,1MB ανά δευτερόλεπτο.



**Εικόνα 50 – Εκκίνηση Εγκατάστασης**

Εφόσον όλα πάνε καλά, το σύστημα θα μας εμφανίσει το μήνυμα πως το λειτουργικό σύστημα (ή τα λειτουργικά συστήματα, αν επιλέξαμε πολλαπλά) εγκαταστάθηκαν επιτυχώς.



**Εικόνα 51 – Επιτυχής Εγκατάσταση**

Κάνοντας κλικ στο OK, το Raspberry Pi θα κάνει επανεκκίνηση.



## Εγκατάσταση OpenCV

Για την εγκατάσταση του πλαισίου εφαρμογής OpenCV είναι απαραίτητη η εκτέλεση των παρακάτω εντολών στο τερματικό(terminal) του raspberry pi.

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

```
$ sudo rpi-update
```

Εγκατάσταση των εργαλείων προγραμματιστή

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

Εγκατάσταση των απαραίτητων πακέτων εισόδου/εξόδου (image libraries).

```
$ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
```

Εγκατάσταση της βιβλιοθήκης GTK.

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

Εγκατάσταση των απαραίτητων πακέτων εισόδου/εξόδου (video libraries).

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

Εγκατάσταση των απαραίτητων πακέτων οι οποίες είναι απαραίτητες για την βελτιστοποίηση της βιβλιοθήκης opencv.

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

Εγκατάσταση του pip.

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

```
$ sudo python get-pip.py
```

Εγκατάσταση του εικονικού περιβάλλοντος virtualenv και virtualenvwrapper :

```
$ sudo pip install virtualenv virtualenvwrapper
```

```
$ sudo rm -rf ~/.cache/pip
```

```
$ source ~/.profile
```

```
$ mkvirtualenv cv
```

Εγκατάσταση των εργαλείων προγραμματισμού της python 2.7.

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

Εγκατάσταση της βιβλιοθήκης Numpy. Το opencv χρησιμοποιεί πίνακες numpy για να την αναπαράσταση εικόνων.

```
$ pip install numpy
```

Μεταφόρτωση του κώδικα της βιβλιοθήκης του opencv, την μεταγλώττιση του και τη δημιουργία εκτελέσιμου.

```
$ 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
```

```
$ 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 ..
```

```
$ make
```

```
$ 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
```

Έλεγχος ότι η βιβλιοθήκη opencv έχει εγκατασταθεί.

```
$ workon cv
```

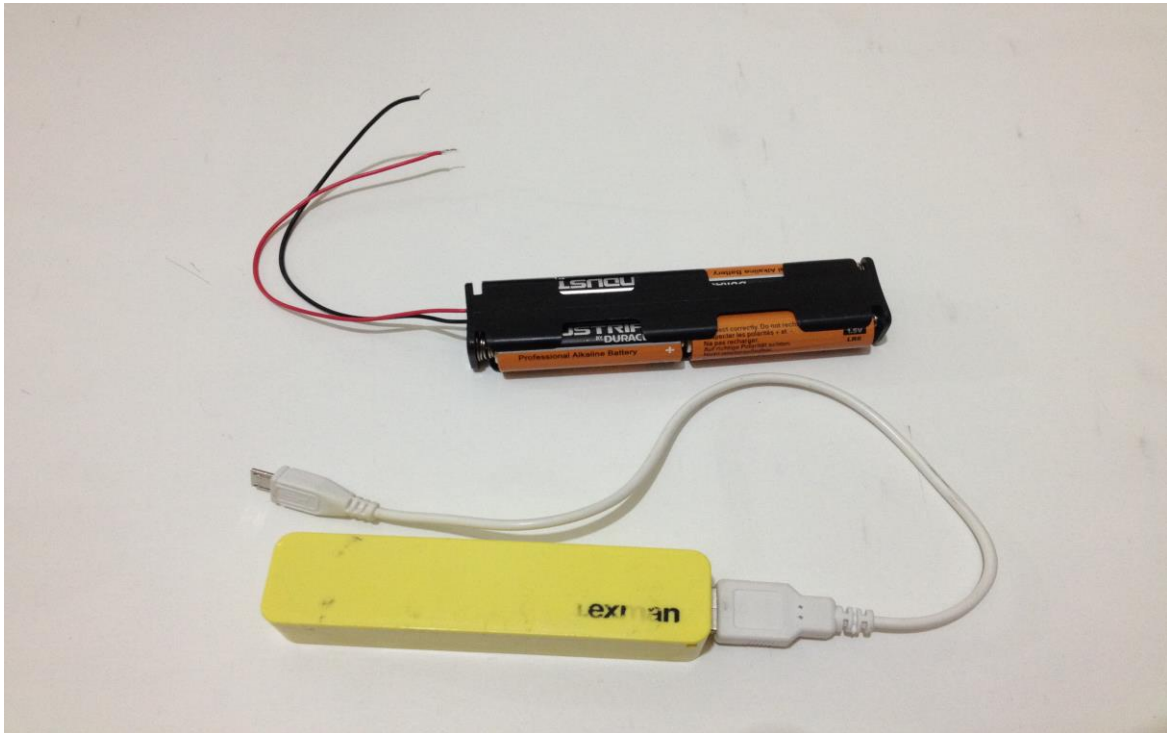
```
$ python
```

```
>>> import cv2
```

```
>>> cv2.__version__
```

```
'2.4.10'
```

## Power Bank



Εικόνα 52 – Power Bank

Τα Power bank τα χρησιμοποιούμε για να είναι το οχημά μας αυτόνομο και να μην χρειαστεί να συνδέσουμε καλώδια για να του δώσουμε ρεύμα. Το κίτρινο power bank το χρησιμοποιούμε για να δώσουμε ρεύμα στο raspberry pi, ενώ το μαύρο για να δώσουμε στο κύκλωμα του breadboard.

## Κώδικας

### **led.py**

(Το led.py είναι πρόγραμμα το οποίο ανάβει ένα φωτάκι led για 10 δευτερόλεπτα και μετά σβήνει).

```
import RPi.GPIO as GPIO

from time import sleep

GPIO.setmode(GPIO.BOARD)

GPIO.setup(25,GPIO.OUT)

GPIO.output(25,1)

sleep(10)

GPIO.output(25,0)

sleep(10)

GPIO.cleanup()
```

### **dcmotor.py**

(Το dcmotor.py είναι πρόγραμμα το οποίο ελέγχει τη λειτουργία του οχήματος και των dc κινητήρων σύμφωνα με τις εντολές που τυπώνονται στην οθόνη δηλαδή αριστερά, δεξιά ή ευθεία).

```
import RPi.GPIO as GPIO

from time import sleep

def set(property, value):

    try:

        f = open("/sys/class/rpi-pwm/pwm0" + property,'w')

        f.write(value)

        f.close()

    except:

        print("error")
```

```
def init():  
    GPIO.setmode(GPIO.BOARD)  
    GPIO.setup(16, GPIO.OUT)  
    GPIO.setup(18, GPIO.OUT)  
    GPIO.setup(22, GPIO.OUT)  
  
    GPIO.setup(19, GPIO.OUT)  
    GPIO.setup(21, GPIO.OUT)  
    GPIO.setup(23, GPIO.OUT)  
    sleep(2)
```

```
def right():  
    print "right"  
    GPIO.output(19, GPIO.LOW)  
    GPIO.output(21, GPIO.HIGH)  
    GPIO.output(23, GPIO.HIGH)
```

```
def left():  
    print "left"  
    GPIO.output(19, GPIO.HIGH)  
    GPIO.output(21, GPIO.LOW)  
    GPIO.output(23, GPIO.HIGH)
```

```
def stop():  
    print "turn motors off"  
    GPIO.output(22, 0)  
    GPIO.output(16, 0)  
    GPIO.output(19, 0)
```

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

```
GPIO.output(23, 0)
sleep(3)

def go():
    print "go"
    GPIO.output(22, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(16, GPIO.HIGH)
init()
go()
sleep(2)

right()
sleep(2)

left()
sleep(2)

go()
sleep(2)

stop()
GPIO.cleanup()
```

### **linedetect.py**

(Το linedetect.py είναι το βασικό πρόγραμμα της εργασίας καθώς τραβάει βίντεο, επεξεργάζεται την εικόνα και ακολουθεί την κόκκινη γραμμή σύμφωνα με την διαδρομή της πίστας).

```
import numpy as np

import cv2

import RPi.GPIO as GPIO

from time import sleep

import time

def init():

    GPIO.setmode(GPIO.BOARD)

    GPIO.setup(16, GPIO.OUT)

    GPIO.setup(18, GPIO.OUT)

    GPIO.setup(22, GPIO.OUT)

    GPIO.setup(19, GPIO.OUT)

    GPIO.setup(21, GPIO.OUT)

    GPIO.setup(23, GPIO.OUT)

    sleep(2)

def right():

    GPIO.output(19, GPIO.LOW)

    GPIO.output(21, GPIO.HIGH)

    GPIO.output(23, GPIO.HIGH)

def stopright():

    GPIO.output(21,0)

    GPIO.output(23,0)
```

```
def left():  
    GPIO.output(19, GPIO.HIGH)  
    GPIO.output(21, GPIO.LOW)  
    GPIO.output(23, GPIO.HIGH)
```

```
def stopleft():  
    GPIO.output(19,0)  
    GPIO.output(23,0)
```

```
def stop():  
    GPIO.output(22, 0)  
    GPIO.output(16, 0)  
    GPIO.output(19, 0)  
    GPIO.output(23, 0)  
    GPIO.output(21, 0)
```

```
def go():  
    GPIO.output(22, GPIO.HIGH)  
    GPIO.output(18, GPIO.LOW)  
    GPIO.output(16, GPIO.HIGH)  
    pwm.start(50)
```

```
cxOld, cyOld = None, None
```

```
cap = cv2.VideoCapture(0)
```

```
cap.set(3,600)
```

```
cap.set(4,400)
```



## Κίνηση Οχήματος με Raspberry Pi και Line Camera

```
init()

sleep(2)

pwm = GPIO.PWM(16,15)
sleep(2)

timeout = time.time() + 30
while (True):
    if time.time() > timeout:
        break

    ret, image = cap.read()

    blur = cv2.GaussianBlur(image, (5,5), 0)

    hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)

    lower_red = np.array([0,100,100])
    upper_red = np.array([20,255,255])

    mask = cv2.inRange(hsv, lower_red, upper_red)

    bmask = cv2.GaussianBlur(mask, (5,5), 0)

    moments =cv2.moments(bmask)

    m00 = moments['m00']
```

```
cx, cy =None, None
```

```
if m00 != 0:
```

```
    cx = int(moments['m10']/m00)
```

```
    cy = int(moments['m01']/m00)
```

```
    ctr = (-1,1)
```

```
if cx != None and cy != None:
```

```
    ctr = (cx, cy)
```

```
    cv2.circle(image,ctr,10,(10,10,10))
```

```
        print "go"
```

```
        go()
```

```
if cxOld == None and cyOld == None:
```

```
    cxOld = cx
```

```
    cyOld = cy
```

```
if cy != None and cx != None and cxOld != None and cyOld !=None:
```

```
    print cx, cy
```

```
    print cxOld, cyOld
```

```
    if (cxOld-cx) > 10:
```

```
        print "move left"
```

```
        left()
```

```
        sleep(0.1)
```

```
        go()
```

```
    if (cx-cxOld) > 10:
```

```
        print "move right"
```

```
        right()
```

## Κίνηση Οχήματος με Raspberry Pi και Line Camera

```
sleep(0.1)
go()

if (cxOld-cx < 10 and cx-cxOld < 10):
    print "go"
    go()
cxOld = cx
cyOld = cy

cv2.imshow('frame', image)
if cv2.waitKey(1) & 0xff == ord('q'):
    stop()
    break

print "stop all"
stop()
pwm.stop()
cap.release()
cv2.destroyAllWindows()
stopleft()
stopright()
sleep(2)
```

(Πιο αναλυτικά σχόλια για τους κώδικές υπάρχουν σε προηγούμενα κεφάλαια καθώς και στο cd της πτυχιακής).

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### 1. Αναφορά σε άλλη εργασία (Διπλωματική - Πτυχιακή, Ερευνητική)

[1] - ΤΖΑΦΕΣΤΑΣ, Σ.Γ., Αυτοματισμός: Η Μεγάλη Επανάσταση, Φυσικός Κόσμος, ΕΕΦ, Φεβρ.1970.

[2] - [http://courseware.mech.ntua.gr/ml23419/robotics\\_pdf/lecture\\_1\\_kk.pdf](http://courseware.mech.ntua.gr/ml23419/robotics_pdf/lecture_1_kk.pdf)

[4] - «Κατασκευή και Πλοήγηση Ρομπότ Ακουστικών Μετρήσεων», Δημοσθένης Ρωσσίδης, Γεώργιος Καμπουράκης, Σεπτέμβριος 2012

[6] - [users.uoa.gr/~akolonou/image\\_processing.pdf](http://users.uoa.gr/~akolonou/image_processing.pdf)

[7] - [www.intelligence.tuc.gr/lib/downloadfile.php?id=303](http://www.intelligence.tuc.gr/lib/downloadfile.php?id=303)

[10]-[nemertes.lis.upatras.gr/jspui/bitstream/10889/1650/1/Diplwmatiki\\_Ergasia.pdf](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/1650/1/Diplwmatiki_Ergasia.pdf)

### 2. Αναφορά σε άρθρο

[3] - SCOTT,P.B, The Robotics Revolution, Basil Blackwell, 1984.

### 3. Αναφορά σε ιστοσελίδα

[5] - <https://sites.google.com/site/epheureseis/inventions/odometro>

[8] - <https://www.python.org/>

[9] - [https://en.wikipedia.org/wiki/Gaussian\\_blur](https://en.wikipedia.org/wiki/Gaussian_blur)

[11] - <https://www.raspberrypi.org/>

