



ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ ΕΥΕΛΠΙΔΩΝ  
Τμήμα Στρατιωτικών Επιστημών

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΔΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2016-17  
ΣΧΕΔΙΑΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ

ΣΥΣΤΗΜΑΤΩΝ (SYSTEMS ENGINEERING)

(ΠΔ 96 /2015/ΦΕΚ 163Α'/20.08.2014)



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
Σχολή Μηχανικών Παραγωγής & Διοίκησης

# ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Εντοπιστής Στόχου – Κατευθυντής

Βλήματος «TheRaven»

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων  
για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

Υπό:

Χρήστου Βαζούρα

A.M.: 2016018010

ΦΕΒΡΟΥΑΡΙΟΣ 2019



**ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

Καθηγητής ΟΝΟΜΑΤΕΠΩΝΥΜΟ κ. Τσαφάρáκης Στέλιος

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

© Copyright υπό .....

Έτος 2019

*Αφιέρωση,*

*Η προσφορά όλου του ακαδημαϊκού περιβάλλοντος, έμψυχο και άψυχο δυναμικό, ήταν ανεκτίμητη, όμως, είμαι ηθικά υποχρεωμένος να αφιερώσω αβίαστα την επίτευξη του προσωπικού μου αυτού στόχου στους ανθρώπους που καθημερινά μου δίνουν το χώρο, το χρόνο και πρωτίστως, το κίνητρο, να συνεχίζω και να προσπαθώ, και δεν είναι άλλοι από τη σύζυγό μου Ατματζίδου Πασχαλίνα και τους τρεις υιούς μας...*

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

*Ευχαριστίες,*

*Στη Στρατιωτική Σχολή Ευελπίδων για τον σχεδιασμό, οργάνωση και διεξαγωγή του εν λόγω Προγράμματος Μεταπτυχιακών Σπουδών, για τη συνεχή υποστήριξη και εξυπηρέτηση των φοιτητών και την διαρκή προσπάθεια για βελτιστοποίηση των δομών και υποδομών του Προγράμματος.*

*Στο Πολυτεχνείο Κρήτης για τον σχεδιασμό, οργάνωση και διεξαγωγή του εν λόγω Προγράμματος Μεταπτυχιακών Σπουδών, για την παροχή της πολυετούς εμπειρίας των καθηγητών του στο χώρο και την διάθεση του εκπαιδευτικού υλικού στους φοιτητές.*

*Στον επιβλέπων την Μεταπτυχιακή μου Διατριβή Καθηγητή, Σχη (ΕΠ) Φραγκουλόπουλο Εμμανουήλ, για την άριστη συνεργασία και κατεύθυνσή του, υποβοηθώντας με έτσι, να δημιουργήσω το “TheRaven”.*

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ.....	1
ΚΕΦΑΛΑΙΟ 1	
Καθορισμός στόχου	
§1. Ανάλυση Ανατεθέντος Έργου.....	3
§2. Παρούσα Κατάσταση – Υπάρχουσα, Χρησιμοποιούμενη Τεχνολογία	4
§3. Πιθανές Εφαρμογές.....	4
§4. Κατασκευή – Φυσική Αρχιτεκτονική.....	5
§5. Περιορισμοί.....	6
§6. Καθορισμός Τελικού Έργου.....	6
ΚΕΦΑΛΑΙΟ 2	
Πυρομαχικά	
§1. Εισαγωγή.....	9
§2. Είδη Πυρομαχικών.....	9
§3. Βλήματα Πυροβολικού.....	10
3.1 Προωθητικές Γομώσεις.....	10
3.2 Βλήματα.....	11
§4. Πύραυλοι.....	14
4.1 A/T Kornet.....	15
4.2 A/T Fagot.....	16
4.3 A/T Javelin.....	17
4.4 A/T Strim.....	17
ΚΕΦΑΛΑΙΟ 3	
Hardware Analysis	
§1. Εισαγωγή.....	19
§2. RaspberryPi ModelB+.....	19

§3. RaspberryPi Camera Module.....	22
§4. Servo Motor Tower Pro MG996R.....	23
§5. RaspberryPi Case.....	24
§6. Λοιπά Υλικά Κατασκευής.....	24

## ΚΕΦΑΛΑΙΟ 4

### Βήματα της Κατασκευής

§1. Εισαγωγή.....	28
§2. Κατασκευή του Κυρίως Μέρους.....	28
§3. Κατασκευή του Φορέα – Βάσης Στήριξης.....	29

## ΚΕΦΑΛΑΙΟ 5

### Software Analysis

§1. Εισαγωγή .....	31
§2. Λειτουργικό Σύστημα.....	31
§3. Γλώσσες Προγραμματισμού.....	34
3.1 C.....	34
3.2 C++.....	35
3.3 Objective C.....	35
3.4 C# .....	35
3.5 Java.....	36
3.6 Javascript.....	36
3.7 Swift.....	36
3.8 PHP.....	37
3.9 Ruby.....	37
3.10 Python.....	37
§4. Επιλογή Γλώσσας Προγραμματισμού.....	38
§5. Βιβλιοθήκες Νευρωνικών Δικτύων.....	40
5.1 Open Source Computer Vision Library.....	41
5.2 Keras.....	42

§6. Επιλογή Βιβλιοθήκης Νευρωνικών Δικτύων .....	43
§7. Ολοκληρωμένο Λογισμικό.....	44
§8. Επιμέρους Ενέργειες.....	46
§9. Ανάλυση Κώδικα.....	47

## ΚΕΦΑΛΑΙΟ 6

### Υλοποίηση Έργου

§1. Κατάσταση.....	53
§2. Ποσοστό Ικανοποίησης Τεθέντος Έργου.....	53
2.1 Hardware.....	53
2.2 Software.....	54
2.3 DIY Κατασκευή.....	54
§3. Δοκιμές.....	54
3.1 Φωτισμός.....	55
3.2 Στόχος.....	55
3.3 Απόσταση.....	56

## ΚΕΦΑΛΑΙΟ 7

### What Else? What more?

§1. Εναλλακτική Τεχνολογία – Προοπτική.....	57
§2. Χρονικές Λεπτομέρειες Έργου.....	60

ΕΠΙΛΟΓΟΣ.....	61
---------------	----

ΒΙΒΛΙΟΓΡΑΦΙΑ.....	62
-------------------	----



## ΕΙΣΑΓΩΓΗ

Η επιλογή του τομέα του θέματος ξεκίνησε περί τα τέλη του Γ' εξαμήνου, την περίοδο του Δεκέμβρη 2017, όταν και αποφάσισα ότι δεν επιθυμώ να διεκπεραιώσω μία πλούσια εργασία σε γραπτό λόγο και θεωρητικό επίπεδο. Ενδιαφέρθηκα από την πρώτη στιγμή, για κατασκευή κάποιου πρωτοτύπου – λογισμικού. Μετά από την συνομιλία με τον Δρ. Φραγκουλόπουλο Εμμανουήλ είχα αποφασίσει ότι το θέμα μου θα ήταν η ανάπτυξη ενός πρωτοτύπου με βάση το Raspberri Pi 3 και θα είχε πιθανή στρατιωτική εφαρμογή. Μετά από πολλαπλές συζητήσεις με τον επιβλέποντα καθηγητή, συγκεκριμενοποιήθηκε η παραπάνω ιδέα και άρχισε να παίρνει σάρκα και οστά με την προμήθεια του hardware. Παρακάτω φαίνεται αναλυτικά η δομή της κατασκευής που ακούει στο όνομα “The Raven” [Εικ.1], βήμα – βήμα, μέχρι την περάτωσή της, όνομα το οποίο παραπέμπει στην αποτελεσματικότητα της επιτυχίας, της πανουργίας αλλά και της «ομορφιάς» του εν λόγω πτηνού.



(Εικ.1)

Ένα σύστημα σαν το TheRaven μπορεί να έχει πολλαπλές εφαρμογές στον τομέα της αμυντικής βιομηχανίας. Η πιο εφικτή και προσιτή περίπτωση είναι να προσαρτηθεί στο σώμα ενός βλήματος καμπύλης τροχιάς ή ενός πυραύλου και να τροποποιεί την πορεία της πτήσης του, με σκοπό να το κατευθύνει στο στόχο.

Η βιομηχανία Αμυντικών Συστημάτων είναι ένας κλάδος που βαίνει ολοένα και αναπτυσσόμενος σε παγκόσμια κλίμακα, ακόμη και από χώρες που πριν 50 χρόνια θα φάνταζε απίστευτο να έχουν τόσο υψηλή παραγωγή στην οπλοβιομηχανία. Στα μέσα του 20 αιώνα η Ρωσία και οι Η.Π.Α. ήταν οι πρωτοστάτες στην παραγωγή όπλων, οπλικών συστημάτων, πυραυλικών συστημάτων, στρατιωτικών οχημάτων, αεροσκαφών, πλοίων, υποβρυχίων και πολλά άλλα. Σε τέτοιο βαθμό που ακόμη και οι πανίσχυροι οικονομικά της εποχής προμηθεύονταν τα οπλικά τους συστήματα από τους παραπάνω. Με την πάροδο των δεκαετιών, βλέπουμε και άλλες δυνάμεις να εισέρχονται στο χώρο όπως την Ινδία, την Κίνα ακόμη και την Κορέα και την γείτονα Τουρκία. Μάλιστα, έχουνε φτάσει στο σημείο εκτός από την αυτάρκεια που έχουν αποκτήσει κάποια κράτη, να έχουν καταφέρει να εξοπλίζουν και άλλα κράτη επιφέροντας έτσι μεγάλη ανάπτυξη στην οικονομία αλλά και στον τομέα της αμυντικής βιομηχανίας.

Η Ελλάδα αντίθετα, δεν έχει παρουσιάσει κάποια παρόμοια πορεία, ούτε στο ελάχιστο, εξακολουθώντας να προμηθεύεται την πλειοψηφία των υλικών των Ενόπλων Δυνάμεων από ξένα κράτη και εταιρείες. Με ελάχιστες εξαιρέσεις που αφορούν κυρίως οχήματα, απάρτια οπλικών συστημάτων, οπτικά μέσα αλλά και επισκευές πολλών συστημάτων, έχει αποκτήσει η χώρα μας μία σχετική αυτάρκεια. Την τελευταία δεκαετία της οικονομικής ύφεσης γίνονται πολλές προσπάθειες προς αυτήν την κατεύθυνση, έχοντας τις περισσότερες φορές θετικά αποτελέσματα, δείχνοντας ότι υπάρχει ευοίωνο μέλλον.

./.

## ΚΕΦΑΛΑΙΟ 1

### Καθορισμός Στόχου

#### 1. Ανάλυση Ανατεθέντος Έργου

Το ανατεθέν θέμα εργασίας «Ενοπιστής Στόχου – Κατευθυντής Βλήματος TheRaven», αφορά σε κάποιο τμήμα ενός οπλικού συστήματος, το οποίο θα εκτελεί κάποιες διεργασίες και θα βελτιώνει την αποδοτικότητα του συστήματος. Πιο συγκεκριμένα, θα πρέπει να έχει τη δυνατότητα να «βλέπει» ούτως ώστε να μπορεί να εντοπίσει κάποιον πιθανό στόχο. Έπειτα, πρέπει να «αναγνωρίσει» το στόχο για να τροποποιήσει την κατεύθυνση του βλήματος και τέλος, να «κατευθύνει» το βλήμα πάνω στο στόχο. Προφανώς, κάποιος πρέπει να «εγκρίνει» και να «δίνει εντολές» στο συγκεκριμένο σύστημα, γεγονός που δεν δύναται να το κάνει ο άνθρωπος, καθόσον τα χρονικά περιθώρια αντίδρασης σε πολλές περιπτώσεις είναι ιδιαίτερος περιορισμένα.

Επομένως, όπως προκύπτει από την παραπάνω ανάλυση, θα απαιτηθούν για την κατασκευή του TheRaven, τα κάτωθι:

- α. Μία κάμερα, η οποία θα ανιχνεύει το οπτικό πεδίο.
- β. Ένας επεξεργαστής, ο οποίος θα εγκρίνει και θα «κλειδώνει» το στόχο, αλλά και θα δίνει τις κατάλληλες εντολές για αλλαγή της πορείας του βλήματος.
- γ. Ένας σερβομηχανισμός, ο οποίος θα εκτελεί την εντολή του επεξεργαστή και θα κινείται μεταβάλλοντας έτσι την πορεία του οπλικού συστήματος.

Μία πλατφόρμα που δύναται να εκτελέσει όλους τους παραπάνω στόχους είναι ο υπολογιστής RaspberryPi ο οποίος, δύναται να δεχτεί πολλά διαφορετικά είδη περιφερειακών και να κατασκευαστεί το λογισμικό του από την αρχή. Έτσι, υπάρχει η δυνατότητα να κατασκευαστεί το ανατεθέν έργο εκ του μηδενός, εύκολη πρόσβαση στα περιφερειακά,

μεγάλη ποικιλία από δυνατότητες σε επίπεδο λογισμικού και με την κατασκευή αντίστοιχης ‘προέκτασης’ να εκμεταλλευόμαστε τις αντίστοιχες διεργασίες του λογισμικού μας.

## 2. Παρούσα Κατάσταση – Υπάρχουσα Χρησιμοποιούμενη Τεχνολογία

Στις μέρες μας, υπάρχουν πολλές διαφορετικές τεχνολογίες που χρησιμοποιούνται παγκοσμίως στην κατεύθυνση και καθοδήγηση πυραύλων, βλημάτων ή λοιπών όπλων μεγάλου βεληνικού. Είναι ένα πολύ σημαντικό τμήμα της Βιομηχανίας Άμυνας, καθόσον η εξέλιξη έγκειται στην ολοένα και μεγαλύτερη απόσταση του ανθρώπινου δυναμικού από τις στρατιωτικές επιχειρήσεις, με σκοπό την ασφάλεια και τις λιγότερες δυνατές απώλειες.

Ένα είδος αυτόνομης κατεύθυνσης είναι τα πυραυλικά συστήματα τα οποία «έλκονται» από το χρώμα και την φωτεινότητα του στόχου, εξετάζοντας και κρίνοντας το μήκος κύματος του χρώματος, όπως π.χ. AGM-84H/K SLAM-ER.

Επίσης, πρέπει να αναφερθούν τα συστήματα πυραύλων ή βλημάτων που με τηλεχειρισμό, μέσω σύρματος από το όπλο ως το βλήμα, μεταβάλλει ο σκοπευτής την πορεία του, ούτως ώστε να πετύχει το στόχο του (π.χ. ο anti – tank πύραυλος Milan).

Πιο εξελιγμένα συστήματα διαθέτουν ραντάρ, το οποίο με αντίστοιχη υποστήριξη ηλεκτρονικών, καθιστά το οπλικό σύστημα που το διαθέτει παντελώς αυτόνομο και υψηλής ακρίβειας (π.χ. ο anti – ship πύραυλος Harpoon).

Αξίζει να αναφερθεί μία άλλη κατηγορία οπλικών συστημάτων τα οποία με δέκτη λέιζερ λαμβάνουν την ακριβή θέση του στόχου, εφόσον τα χερσαία τμήματα τον καταδεικνύουν με τον αντίστοιχο καταδείκτη λέιζερ, όπως π.χ. οι βόμβες GBU10-12-16 που διαθέτουν τα Α/Φ.

## 3. Πιθανές Εφαρμογές

Το TheRaven θα μπορούσε να σχεδιαστεί – κατασκευαστεί με σκοπό να προσαρμοστεί πάνω σε βλήμα καμπύλης τροχιάς που βάλλεται από πυροβόλο όπλο, γεγονός



που θα έδινε τη δυνατότητα στο υπόψιν οπλικό σύστημα να εκτελεί στοχευμένα πυρά με αισθητά αυξημένη αποτελεσματικότητα.. Αντίστοιχα, ένας πύραυλος που χρησιμοποιεί ένα αεροσκάφος, ένα αντιαερατικό όπλο ή ένα πυραυλικό σύστημα διαφόρων τύπων, θα μπορούσε να διαθέτει το TheRaven με σκοπό την εξασφάλιση της προσβολής του στόχου και την εξασφάλιση της φονικότητας του οπλικού συστήματος. Προφανώς, η κατασκευή του TheRaven θα διαφέρει αναλόγως σε ποιο όπλο – πυρομαχικό θα προσαρτηθεί, εξαιτίας πολλών παραγόντων, όπως η ταχύτητα, το είδος της πορείας που διαγράφει, το σχήμα και το μέγεθος του βλήματος – πυραύλου, τον τρόπο με τον οποίο βάλλεται αλλά και το είδος του στόχου για το οποίο είναι κατασκευασμένο το αντίστοιχο πυρομαχικό.

#### 4. Κατασκευή – Φυσική Αρχιτεκτονική

Η κατασκευή του TheRaven θα απαρτίζεται από δύο σκέλη: το λογισμικό του και την δημιουργία μίας κατασκευής, η οποία θα ελέγχεται από το RaspberryPi και θα έχει τη δυνατότητα να μετατρέπει την πορεία του πυρομαχικού πάνω στο οποίο θα έχει προσαρτηθεί.

Όσον αφορά τον κώδικα που απαιτείται για να καθορίζει επ' ακριβώς τις διεργασίες και τη λειτουργία του TheRaven, θα πραγματοποιηθεί εξ' αρχής, ανατρέχοντας σε διάφορες πηγές του διαδικτύου, οι οποίες προσφέρουν πληθώρα παραδειγμάτων και ιδεών για τις τεχνικές και τους τρόπους με τους οποίους μπορεί κάποιος να εκτελέσει διάφορες διεργασίες. Η γλώσσα προγραμματισμού η οποία έχει πολλές δυνατότητες και επιλογές, αλλά και πολλές απλουστεύσεις σε σχέση με την C++, η οποία θεωρείται 'μητέρα' όλων, είναι η Python. Με την Python μπορούμε να ελέγχουμε πλήρως το TheRaven και να του καθορίσουμε πώς ακριβώς θα ενεργεί σε κάθε περίπτωση.

Η κατασκευή, ωστόσο, θα γίνει από απλά υλικά και θα έχει μία γενική μορφή, και όχι κάτι εξεζητημένο το οποίο θα ταίριαζε σε κάποιο συγκεκριμένο είδος πυρομαχικού, και θα είναι κατασκευασμένο από τελείως απλά υλικά. Έτσι, η κατασκευή μας θα απεικονίζει το

οπίσθιο μέρος ενός βλήματος – πυραύλου, το οποίο θα μεταβάλλει την πορεία του, με τη χρήση δύο πτερυγών, πτέρυγες οι οποίες θα μεταφράζουν τη λειτουργία του TheRaven.

## 5. Περιορισμοί

Η δημιουργία του TheRaven διέπεται από πολλές επιλογές και ποικιλία τρόπων διεκπεραίωσης, αλλά αντίστοιχα, υφίστανται και διάφοροι περιορισμοί, όπως παρακάτω:

✚ Ο RaspberryPi είναι ένας υπολογιστής ο οποίος έχει σχετικά χαμηλή απόδοση και αυτό οφείλεται αποκλειστικά και μόνο στον επεξεργαστή που διαθέτει. Εν ολίγοις, θα εκτελέσει κανονικά όλες τις επιμέρους διεργασίες που θα έχουμε ζητήσει, αλλά με κάποια καθυστέρηση χρόνου.

✚ Η κάμερα που θα χρησιμοποιηθεί και είναι της ομώνυμης εταιρείας είναι μετρίων προδιαγραφών και αυτό προκαλεί δυσχέρεια στην αναγνώριση του στόχου και τον εντοπισμό του από σχετικά μεγάλη απόσταση. Επίσης, ο φωτισμός παίζει καθοριστικό ρόλο στην εύκολη και έγκαιρη αναγνώριση του στόχου από την κάμερα.

✚ Η δυνατότητα του RPi να τροφοδοτήσει κομμάτια Hardware που θα προσαρτηθούν είναι περιορισμένη, καθόσον δύναται να υποστηρίξει μόνο την κάμερα και ένα σερβομηχανισμό. Ακόμη και σ' αυτήν την περίπτωση, υπάρχει πιθανότητα απροειδοποίητης επανεκκίνησης του συστήματός μας, λόγω του γεγονότος 'αστάθειας' της τάσης του ρεύματος.

✚ Δεν υφίσταται πρόσβαση σε κατασκευές βιομηχανικού επιπέδου και υλικά αυξημένων προδιαγραφών, με αποτέλεσμα να χρησιμοποιηθούν κοινές πρώτες ύλες και εργαλεία, τα οποία θα μπορούν να αντικατασταθούν στο μέλλον από τα κατάλληλα.

## 6. Καθορισμός Τελικού Έργου

Κατόπιν των παραπάνω, θα κατασκευαστεί το TheRaven εξ' ολοκλήρου από την αρχή, ακολουθώντας τα παρακάτω βήματα:

✚ Θα γίνει προμήθεια όλου του απαραίτητου υλικού και περιφερειακών, δηλαδή: RaspberryPi Model B+ , PiCamera, Servomotor, θήκη προστασίας, λοιπών υλικών κατασκευής όπως ξυλεία και πτέρυγες. Η προμήθεια θα γίνει από καταστήματα που βρίσκονται στο Νομό Αττικής, που διαθέτουν όλη την γκάμα των εν λόγω υλικών.

✚ Θα εγκατασταθούν τα προγράμματα και το λογισμικό, με σκοπό να εξασφαλιστεί η εύρυθμη λειτουργία του υπολογιστή μας.

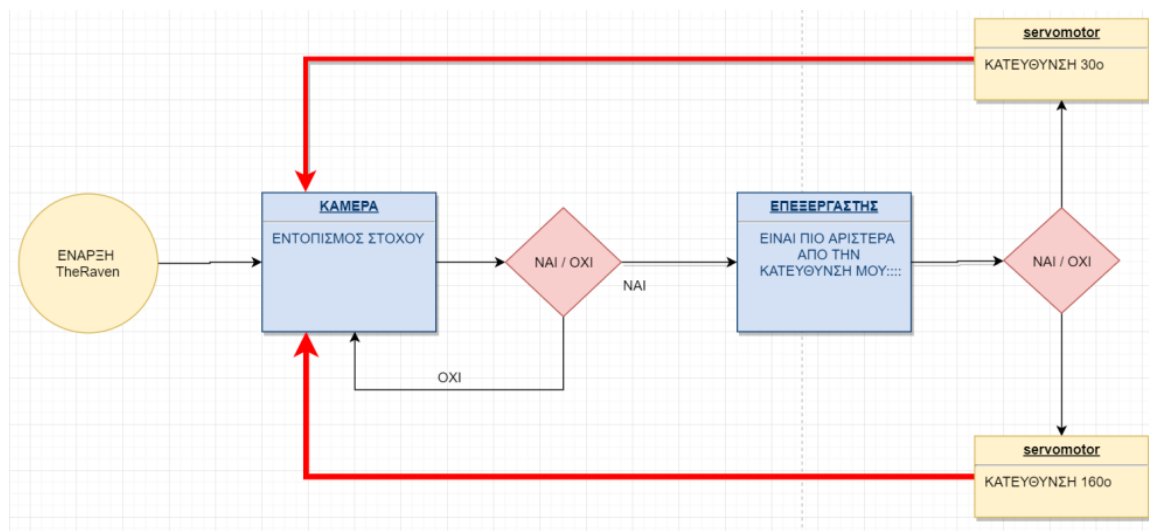
✚ Θα γίνει συγγραφή του αρχείου το οποίο θα εκτελεί τις διεργασίες αναγνώρισης και εντοπισμού του στόχου, αλλά και κίνησης του σερβομηχανισμού.

✚ Θα κατασκευαστεί το οπίσθιο τμήμα του βλήματος – πυραύλου, πάνω στον οποίο θα προσαρμοστεί υποθετικά το TheRaven, ούτως ώστε να ‘μεταφράζεται’ η κίνηση του σερβομηχανισμού σε μεταβολή της κατεύθυνσης του βλήματος.

✚ Θα γίνουν δοκιμές για πρόβλεψη και έγκαιρη αντιμετώπιση τυχόν προβλημάτων.

Όσον αφορά το τρόπο κατά τον οποίο θα λειτουργεί το TheRaven, την αλληλουχία των διεργασιών δηλαδή, απεικονίζονται παρακάτω:

### Διάγραμμα Ροής Λειτουργίας του TheRaven



[Εικ.2]

Όπως φαίνεται παραπάνω, με την έναρξη λειτουργίας του TheRaven, θα τίθεται σε εφαρμογή η ανίχνευση της περιοχής από την κάμερα. Η κάμερα διαρκώς προσπαθεί να εντοπίσει κάποιον από τους στόχους, τους οποίους έχει στο «μυαλό» του ο υπολογιστής, αεροσκάφος ή πλοίο στην προκειμένη περίπτωση. Όταν εντοπιστεί ένας στόχος, γίνεται προσδιορισμός των συντεταγμένων του και εφόσον βρίσκεται στο αριστερό ή δεξί κομμάτι της οθόνης μας, δίνεται η εντολή στο σερβομηχανισμό να κινηθεί αντίστοιχα, ούτως ώστε να κατευθύνει τον πύραυλο – βλήμα πάνω στο στόχο.

./.

## ΚΕΦΑΛΑΙΟ 2







### Είδη Πυρομαχιών

#### 1. Εισαγωγή

Αρχικά πρέπει να γίνει μία αναφορά σε μία γιάμα πυρομαχιών και πυραύλων πάνω στα οποία, υπάρχει η δυνατότητα, θεωρητικά, να προσαρμοστεί το TheRaven. Θα ερευνήσουμε τα τεχνικά χαρακτηριστικά αυτών, αλλά και την ανατομία τους, ούτως ώστε να εξεταστούν εις βάθος όλες οι δυνατότητες. Επίσης, ο τρόπος με τον οποίο λειτουργεί το οπλικό σύστημα το οποίο εκτοξεύει το βλήμα – πύραυλο, είναι πολύ σημαντικός, αφού μπορεί να αποτελέσει ανασταλτικό παράγοντα για τη λειτουργία του TheRaven.

#### 2. Είδη πυρομαχιών

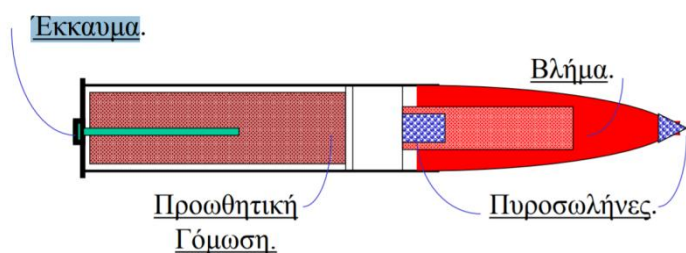
Σαν πυρομαχικά ορίζονται όλα τα βλήματα, προωθητικά γεμίσματα και πάσης φύσεως στρατιωτικών εκρηκτικών υλών που είναι συσκευασμένες, έτοιμες για μετακίνηση και χρησιμοποιούνται για τη διεξαγωγή ενός πολέμου. Στην παγκόσμια βιομηχανία άμυνας συναντώνται πολλά και διάφορα είδη βλημάτων που βάλλονται από διαφορετικά είδη όπλων και μηχανημάτων. Επιγραμματικά, τα είδη των βλημάτων που υφίστανται είναι τα κάτωθι:

-  Πυροβολικού
-  Υφάλων όπλων
-  Αεροφερόμενα
-  Αυτοπροωθούμενα
-  Υπονομεύσεων
-  Χημικά

Παρακάτω θα ακολουθήσει μία ανάλυση σε κάθε είδος των παραπάνω κατηγοριών βλημάτων, ούτως ώστε να δημιουργηθεί μία σφαιρική άποψη για τις πιθανές εφαρμογές του TheRaven.

### 3. Βλήματα Πυροβολικού

Όλα τα είδη βλημάτων Πυροβολικού αποτελούνται από διάφορα μέρη τα οποία ποικίλουν από βλήμα σε βλήμα. Αυτά είναι: η προωθητική γόμωση, το έκκαυμα, το βλήμα και ο πυροσωλήνας.[Εικ.3]



4

[Εικ.3]

Οι παράμετροι ταξινόμησης των βλημάτων Πυροβολικού είναι:

- + Το διαμέτρημα
- + Ο τρόπος συναρμολόγησης – συνεργασίας
- + Αναλόγως της χρήσης τους και για ποιον σκοπό προορίζονται.

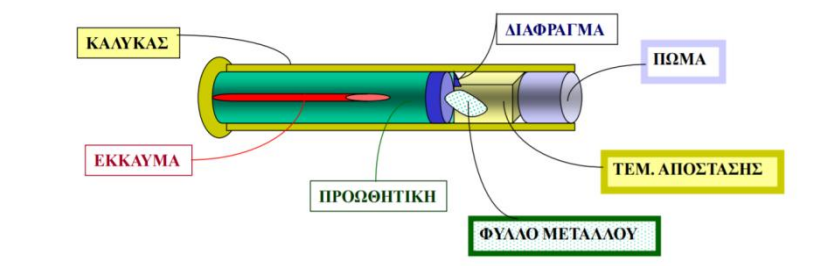
Όσον αφορά τα διαθέσιμα διαμετρήματα τόσο στην Δυτική όσο και στην Ανατολική βιομηχανία Άμυνας απαντώνται διάφορα διαμετρήματα, όπου διαμέτρημα εννοείται η διάμετρος του οπλομηχανήματος από το οποίο βάλλεται το κάθε είδος πυρομαχικού. Έτσι, έχουμε 76 χιλ, 105 χιλ, 155 χιλ, 203 χιλ τα οποία είναι τα ευρέως χρησιμοποιούμενα πυρομαχικά, χωρίς να αποτελούν και αποκλειστικότητα, καθόσον στην Ανατολική Τεχνολογία Άμυνας συναντώνται πολλές εξαιρέσεις και μοναδικές κατηγορίες.

#### 3.1 Προωθητικές Γομώσεις

##### 1/ Τύπος φυσίγγης

Τα μέρη της φυσίγγης είναι τα κάτωθι [Εικ.4]:

- + Έκκαυμα
- + Κάλυκας
- + Προωθητική
- + Διάφραγμα
- + Τεμάχιο απόστασης
- + Φύλλο μετάλλου
- + Πώμα



[Εικ.4]

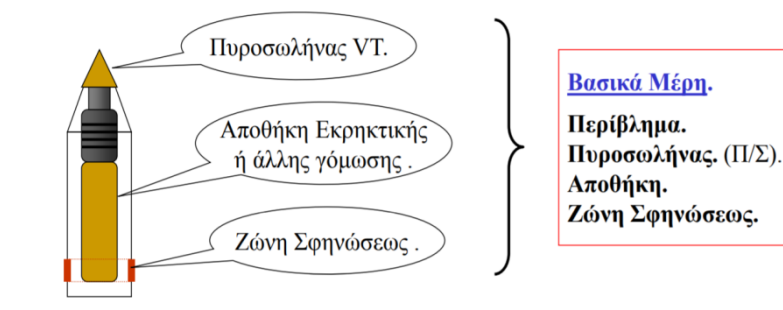
## 2/ Τύπος σακιδίων

Τα σακίδια προωθητικής γόμωσης είναι μικρά υφασμάτινα σακίδια, που περιέχουν συγκεκριμένη ποσότητα προωθητικής γόμωσης και επομένως επιτυγχάνουν συγκεκριμένο βεληνεκές στο βλήμα. Αναλόγως του επιθυμητού βεληνεκούς χρησιμοποιείται και ανάλογος αριθμός σακιδίων.

## 3.2 Βλήματα

Ένας άλλος τύπος πυρομαχικού Πυροβολικού, που χρησιμοποιείται ευρέως στα μεγαλύτερα διαμετρήματα (120χιλ. και άνω) είναι αυτός του ολομερούς βλήματος. Γίνεται λόγος για μία συμπαγή κατασκευή, η οποία περιέχει όλα τα απαραίτητα στοιχεία για την

πυροδότησή του, δηλαδή το περίβλημα, τον πυροσωλήνα, την αποθήκη και τη ζώνη σφηνώσεως, σε διάταξη όπως Εικ.5:



[Εικ.5]

Η σχεδίαση πάντα των βλημάτων γίνεται βάσει κάποιων κριτηριων. Ένα από αυτά είναι το επιθυμητό αποτέλεσμα σε σχέση με το στόχο, εάν δηλαδή επιδιώκεται η ολοσχερής καταστροφή του, η παρεμπόδισή του, εάν προορίζεται μόνο για προσωπικό ή τεθωρακισμένα οχήματα, κτλ. Επίσης, η ανάγκη εξασφάλισης της ευστάθειας κατά την κίνηση είναι ένας άλλος παράγοντας, όπως και η προσπάθεια περιορισμού της αντίστασης – τριβής κατά την κίνηση παράγοντας ο οποίος καθορίζει το μέγιστο βεληνειές του βλήματος.

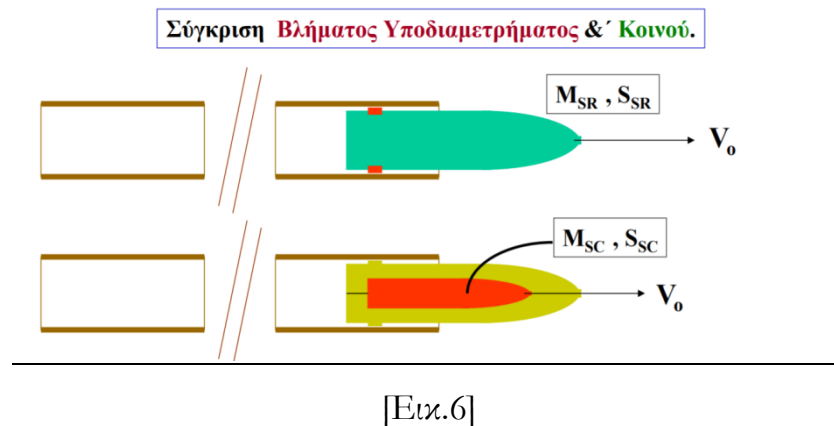
1/ Συμπερασματικά, καταλήγουμε στους εξής τύπους βλημάτων:

- ✚ Κοινά
- ✚ Υποδιαμετρήματος
- ✚ Βοηθητικής πυραυλώθησης
- ✚ Ροής Βάσης (Base – Bleed)

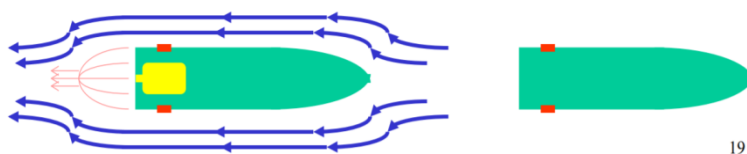
Όσον αφορά τα βλήματα υποδιαμετρήματος, χρησιμοποιείται μία πλαστική καλύπτρα η οποία προσομοιάζει το μέγεθος – σχήμα του κοινού, και χρησιμοποιείται για την ευκολότερη και οικονομικότερη εκπαίδευση του προσωπικού. Άρα με χρήση μίας πλαστικής καλύπτρας Εσωτερικής Βλητικής, που απορρίπτεται στο στόμιο με την έξοδο από τον σωλήνα, το Βλήμα υποδιαμετρήματος απορροφά όλη την Ενέργεια της Προωθητικής κατά την κίνησή του μέσα στον μεγαλύτερης διαμέτρου σωλήνα του Πυροβόλου. Το



πλεονέκτημά του έγκειται εις το ότι η Αρχική Ταχύτητά τού - συγκρίσιμη με αυτήν ενός ίσου βάρους κοινού βλήματος, (Standard round) - θα απομειώνεται με βραδύτερο ρυθμό, λόγω τής μικρότερης τιμής της Οπισθέλκουσας που θα επιδρά σ' αυτό, αφού και η διάμετρος της ορθής τομής τού θα είναι μικρότερη έναντι αυτής τού κοινού βλήματος. Παρακάτω φαίνεται και η διάφορα στην κατασκευή [Εικ.6]:

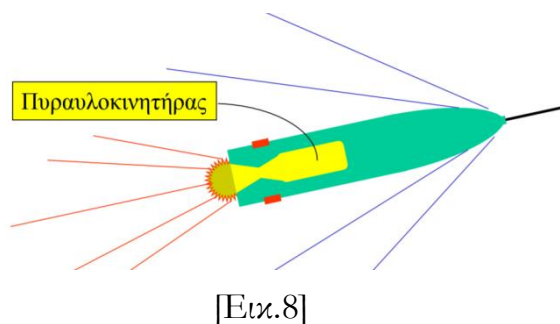


Ακολουθως, η διαφορά μεταξύ κοινού βλήματος και του βλήματος ροής βάσης, έγκειται στο γεγονός ότι επιτυγχάνουμε μεγαλύτερο βεληνεκές. Αυτό συμβαίνει με χρήση ενός φιαλιδίου - το οποίο ευρίσκεται στο πυθμένιο και περιέχει αέριο υπό πίεση - το Βλήμα Ροής Βάσης δημιουργεί, μετά την έξοδο από το στόμιο, μία καλά σχεδιασμένη συμμετρική ως προς τον άξονά τού ροή. Το πλεονέκτημά τού είναι ότι με την δημιουργία και διατήρηση κατά την πτήση μίας εξ αερίου "αντικορυφής" στην περιοχή τού πυθμενίου, αποφεύγεται η δημιουργία στροβιλισμών και βελτιώνεται ο Συντελεστής Οπισθέλκουσάς του, έναντι αυτού του κοινού βλήματος, (Standard round). Συνέπεια αυτού είναι: ότι η Αρχική Ταχύτητα θα απομειώνεται με βραδύτερο ρυθμό, λόγω της μικρότερης τιμής της Οπισθέλκουσας που θα επιδρά επάνω στο βλήμα "Base-Bleed", έναντι αυτής τού κοινού βλήματος [Εικ.7]. Δηλαδή:



[Εικ.7]

Τέλος, το βλήμα Βοηθητικής Πυραυλώθησης, [Εικ.8] χρησιμοποιεί ενσωματωμένο πυραυλοκινητήρα, ο οποίος προσδίδει ταχύτητα επιπλέον της βαλλιστικής του βλήματος, προς επίτευξη μακρότερου βεληνεκούς. Αυτό έχει ως αποτέλεσμα την προσβολή στόχων που βρίσκονται σε μεγαλύτερη απόσταση από το στόχο, αλλά η απόκλιση του αποτελέσματος της βολής είναι αυξημένη. Επομένως, τέτοιου τύπου βλήματα χρησιμοποιούνται για βολή ευρείας περιοχής και όχι σημειακού στόχου.



#### 4. Πύραυλοι

Μία άλλη κατηγορία πυρομαχικών που βάζονται από οπλομηχανήματα πολλών και διαφόρων διαμετρημάτων είναι οι πύραυλοι. Όλες οι κατηγορίες πυραύλων έχουν ένα κοινό: επιφέρουν μεγάλη απώλεια στο στόχο. Συγκρινόμενοι με ίδια διαμετρήματα άλλων κατηγοριών πυρομαχικών υπερτερούν κατά πολύ στο μέγεθος της έκρηξης, στην ακρίβεια της βολής και την ευκολία σκόπευσης, αλλά το σημείο στο οποίο μειονεκτούν αισθητά είναι το βεληνεκές. Βέβαια, δεν αναφερόμαστε στους διηπειρωτικούς πυραύλους ή άλλα υπερσύγχρονα οπλικά συστήματα.

Θα αναφερθούν κάποιες κατηγορίες πυραύλων που χρησιμοποιούνται ευρέως από την παγκόσμια βιομηχανία αμυντικών συστημάτων, ούτως ώστε να γίνει αντιληπτό σε ποιες

περιπτώσεις και υπό ποιες συνθήκες δύνανται το «TheRaven» να έχει εφαρμογή και πότε να έχει απόλυτη επιτυχία.

#### 4.1 A/T Kornet [Εικ.14]



[Εικ.14]

Το A/T Kornet είναι ένα από τα πιο πρόσφατα αντιαερατικά όπλα της παγκόσμιας οπλοβιομηχανίας, καθόσον εισήχθη σ' αυτή το 1998 και χρησιμοποιείται ευρέως μέχρι και σήμερα. Είναι ρωσικής κατασκευής και τεχνολογίας και ανήκει στην κατηγορία των «fire and forget» πυραύλων, αφού «κλειδώνει» τον στόχο με το θερμικό του ίχνος. Διαθέτει διάφορους τύπους πυραύλων με εξέχοντες τον θερμοβαρικό πύραυλο, που χρησιμοποιείται εναντίον στόχων με ελαφριά θωράκιση και τον πύραυλο εκτεταμένου βεληνειούς, που δύναται να προσβάλει στόχο ακόμη και σε απόσταση 10 χλμ.! Η καινοτομία του συνίσταται στο γεγονός ότι είναι φορητό, εύκολο στο χειρισμό – εκπαίδευση και ότι χρησιμοποιεί ακτίνα λέιζερ για την στοχοποίηση. Έχει χρησιμοποιηθεί στον Β' Πόλεμο του Κόλπου, στις επιχειρήσεις στο Αφγανιστάν αλλά και στις επιχειρήσεις στη Συρία μέχρι και σήμερα.

#### 4.2 A/T Fagot [Εικ.15]



[Εικ.15]

Το A/T Fagot είναι άλλο ένα οπλικό σύστημα Ρωσικής προέλευσης και κατασκευής, που εισήχθη στα Θέατρα Επιχειρήσεων παγκοσμίως από το 1970 και χρησιμοποιείται μέχρι και σήμερα. Είναι A/T όπλο δεύτερης γενιάς και ο πύραυλός του κατευθύνεται από το χειριστή, αφού εγκαταλείψει το φορέα, μέσω ενός σύρματος – καλωδίου, το οποίο και μεταφέρει τις εντολές του χειριστή διορθώνοντας έτσι την πορεία του μέχρι να βρει τον στόχο του. Το βεληνεκές του αγγίζει τα 4000μ. και απαιτούνται τρεις (3) άνδρες για την υπηρετήσή του. Το πλεονέκτημα όλων των οπλικών συστημάτων που διαθέτουν το συγκεκριμένο σύστημα κατεύθυνσης (SACLOS) είναι η αποτελεσματικότητα, καθόσον μπορεί να φτάσει και το 90%, αναλόγως πάντα την εκπαίδευση και ικανότητα του χειριστή. Αντιθέτως, λόγω πεπαλαιωμένης τεχνολογίας, δεν δύναται να προσομοιάσει τις δυνατότητες ενός σύγχρονου ‘έξυπνου’ βλήματος.

#### 4.3 Α/T Javelin [Εικ.16]



[Εικ.16]

Το Α/T Javelin είναι από τα πιο σύγχρονα οπλικά συστήματα Α/T που χρησιμοποιεί ο Αμερικανικός Στρατός με μέγιστο βεληνεκές τα 4.750μ. Ανήκει στην κατηγορία ‘fire and forget’ και λόγω της φορητότητάς του, σε συνδυασμό με τις ανεβασμένες δυνατότητές του, αποτελεί το τελευταίο απόκτημα στο οπλοστάσιο των πιο προηγμένων ενόπλων δυνάμεων παγκοσμίως. Χρησιμοποιεί αισθητήρες που ‘κλειδώνουν’ το θερμικό ίχνος του στόχου και εφόσον πυροδοτηθεί κατευθύνεται στο στόχο διαγράφοντας πορεία που διαβάλλει την πλειοψηφία των αντιβαλλιστικών όπλων παγκοσμίως. Αξιζει να αναφερθεί ότι η αποτελεσματικότητά του κυμαίνεται μεταξύ 82 – 99%!!!

#### 4.4 Α/T Strim [Εικ.17]



[Εικ.17]

Το Α/T Strim είναι Γαλλικής κατασκευής και προελεύσεως, πρωτο χρησιμοποιήθηκε στις αρχές της δεκαετίας του ’70 και μέχρι και σήμερα συναντάται στις πολεμικές επιχειρήσεις που διεξάγονται παγκοσμίως. Είναι μικρού βεληνεκούς και εξαιρετικά ελαφρύ και αποτελεσματικό. Δεν διαθέτει καμία από τις προαναφερθείσες

τεχνολογίες και η αποτελεσματικότητά του εξαρτάται στο μεγαλύτερο ποσοστό από τον χειριστή – σκοπευτή.

./.

## ΚΕΦΑΛΑΙΟ 3

### Hardware analysis

#### 1. Εισαγωγή

Αναφορικά με τον υπολογιστή που θα αποτελέσει τον εγκέφαλο του εν λόγω εγχειρήματος, επιλέχθηκε το Single Board Computer της RaspberryPi Foundation και συγκεκριμένα η έκδοση Model B+. Αυτό συνέβη καθόσον, ο συγκεκριμένος υπολογιστής είναι προσιτός σε προσβασιμότητα και τιμή, αλλά συνοδεύεται και από πληθώρα περιφερειακών της ομώνυμης εταιρίας. Επίσης, μετά από αυτό ήταν πολύ απλό να επιλεχθούν τα υπόλοιπα περιφερειακά που θα ταιριάζουν με το RPi. Το είδος του κεντρικού υπολογιστή άλλα και των λοιπών περιφερειακών, επηρεάζουν την απόδοση του TheRaven στο σύνολό του, δηλαδή τόσο τον χρόνο εντοπισμού, την αντίδραση στην πιστοποίηση του στόχου, το ποσοστό επιτυχίας αυτής, την εγυρότητα προσδιορισμού των συντεταγμένων, αλλά και την ταχύτητα διαβίβασης της εντολής στο σερβομηχανισμό για μετακίνηση αριστερά ή δεξιά.

#### 2. Raspberry Pi Model B+

Η εταιρεία RaspberryPi Foundation είναι η πρώτη που εισήγαγε το είδος αυτό του Υπολογιστή, ο οποίος φέρει επεξεργαστή και τις κατάλληλες εισόδους - εξόδους μόνο, και δίνει τη δυνατότητα στον απλό χρήστη να τον 'στήσει' όπως επιθυμεί αναλόγως τις ανάγκες του.

Το RaspberryPi [Εικ.23] είναι η μικρότερη έκδοση προσωπικού υπολογιστή και ανήκει στην κατηγορία Single Board Computer, καθόσον αποτελείται μόνο από τα βασικά στοιχεία ενός υπολογιστή. Διατίθενται διαφορετικές εκδόσεις στην αγορά με την πιο πρόσφατη την RaspberryPi Model B+, η οποία και χρησιμοποιήθηκε, οι οποίες διαθέτουν κοινό SoC (System on Chip – combined CPU and GPU) αλλά διαφέρουν στις προεκτάσεις



του διατιθέμενου hardware. Τα τεχνικά χαρακτηριστικά του RaspberryPi Model B+ έχουν όπως παρακάτω:

- + Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1,4 GHz
- + 1GB SDRAM
- + 2.4 Hz and 5 GHz wireless LAN, Bluetooth 4.2
- + Gigabit Ethernet over USB 2.0
- + 40-pin GPIO header
- + Full-size HDMI
- + 4 USB 2.0 ports
- + CSI camera for connecting a RPi camera Module
- + DSI display port for connecting a RPi touchscreen display
- + 4-pole stereo output and composite video port
- + Micro SD port for loading the OS and storing data
- + 5V/2.5A DC power input



[Εικ.23 – RaspberryPi Model B+]



Η ομώνυμη εταιρεία έχει την επιδίωξη να καταστήσει τον προγραμματισμό ένα πολυχρηστικό εργαλείο διατιθέμενο σε όλον τον πληθυσμό που διαθέτει στοιχειώδεις γνώσεις τεχνολογίας, και όχι αποκλειστικά στους προγραμματιστές που διαθέτουν βαθιές τεχνικές γνώσεις και πολυετείς σπουδές και πείρα. Έτσι, καινοτόμησε πριν σχεδόν μία δεκαετία με την κατασκευή και διάθεση του RaspberryPi, καθόσον το κόστος του είναι απολύτως προσιτό και η συμβατότητά του με οποιαδήποτε άλλη συσκευή τεχνολογίας είναι εφικτή και απλή. Επίσης, διατίθενται τόσο από την εταιρεία όσο και από πληθώρα φανατικών χρηστών, ελεύθερα στο διαδίκτυο πακέτα κώδικα αλλά και οδηγίες για οποιαδήποτε χρήση του RPi. Επομένως, η ενασχόληση ενός μεγάλου αριθμού κυρίως του νεαρού πληθυσμού, και όχι μόνο, με διάφορες καθημερινές εφαρμογές τεχνολογίας όλο και αυξάνεται, έχοντας ήδη καταστήσει αναγκαία την κατηγοριοποίηση όλων αυτών των projects στο λεγόμενο IoT (Internet of Things), και η RaspberryPi έπαιξε πρωταρχικό ρόλο, μαζί με λιγοστές ακόμη εταιρείες όπως Arduino, ASUS, κτλ.

Έτσι, η επιλογή του RaspberryPi προσφέρει αξιοπιστία, εύκολη προμήθεια και κατασκευή της διάθρωσης του TheRaven, με μειωμένη βέβαια απόδοση, λόγω προδιαγραφών. Αυτό που είναι σημαντικό για την κατασκευή του TheRaven είναι να πιστοποιηθεί ότι όντως το TheRaven μπορεί να κατευθύνει έναν πύραυλο – βλήμα, ανεξαρτήτως της απόδοσης ή της υψηλής αξιοπιστίας.

Αξίζει να αναφερθούν κάποια στοιχεία για τη λειτουργία των General Purpose Inputs Outputs Pins τα οποία διαθέτει ο υπολογιστής μας. Είναι 40 στον αριθμό, καθένα με ομαδοποιημένα ανά λειτουργία, δηλαδή, τα υπ' αριθμόν 6, 9, 14, 20, 25, 30, 34 και 39 Pins είναι σημεία γείωσης. Οποιοδήποτε module και αν συνδέσουμε στον RPi πρέπει να συνδέεται σε μία γείωση, ανεξάρτητα αν τροφοδοτείται με ρεύμα από τον υπολογιστή μας. Στη συνέχεια, τα Pins 1, 2, 4 και 17 είναι για τροφοδοσία εξωτερικών 'συσκευών' τα οποία όμως πρέπει να έχουν περιορισμένες ανάγκες, καθόσον σε περίπτωση ανεπάρκειας το RPi θα εκτελεί επανεκκίνηση απροειδοποίητα διακόπτοντας κάθε διεργασία!!!

### 3. Raspberry Pi Camera

Η κάμερα είναι το Camera Module της ομώνυμης εταιρίας με ανάλυση 5.0 MP η οποία συνδέεται κατευθείαν στην υποδοχή της, δοθείσας βέβαια της δυνατότητας να χρησιμοποιηθεί οποιαδήποτε άλλη webcam που δύναται να συνδεθεί σε ένα από τα 4 usb ports. Παρακάτω φαίνονται αναλυτικά τα τεχνικά χαρακτηριστικά της:

	<b>Camera Module v1</b>
Net price	\$25
Size	Around 25 × 24 × 9 mm
Weight	3g
Still resolution	5 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90
Linux integration	V4L2 driver available
C programming API	OpenMAX IL and others available
Sensor	OmniVision OV5647
Sensor resolution	2592 × 1944 pixels
Sensor image area	3.76 × 2.74 mm
Pixel size	1.4 μm × 1.4 μm
Optical size	1/4"
Full-frame SLR lens equivalent	35 mm
S/N ratio	36 dB
Dynamic range	67 dB @ 8x gain
Sensitivity	680 mV/lux-sec
Dark current	16 mV/sec @ 60 C
Well capacity	4.3 Ke-
Fixed focus	1 m to infinity
Focal length	3.60 mm +/- 0.01
Horizontal field of view	53.50 +/- 0.13 degrees
Vertical field of view	41.41 +/- 0.11 degrees
Focal ratio (F-Stop)	2.9

Σε περίπτωση που χρησιμοποιούταν κάποια άλλη κάμερα, θα συνδεόταν σε ένα από τα usb ports του υπολογιστή μας. Αυτό θα μας έδινε τη δυνατότητα καλύτερης ανάλυσης και καθαρότερης εικόνας, αλλά θα έπρεπε να έχουμε κάποιες διαφορές στον κώδικά μας.

#### 4. Servo motor Tower Pro MG996R

Ο σερβομηχανισμός [Εικ.24] που χρησιμοποιήθηκε στην κατασκευή του “The Raven” είναι βασικών προδιαγραφών και δυνατοτήτων, όπως παρακάτω:

- + Βάρος: 55 γρ.
- + Ροπή έλξης: 9,4 kgf\*cm (4,8 V)
- + Ταχύτητα λειτουργίας: 0,17 sec/60° (4,8 V)
- + Εύρος τάσης: 4,8 – 7,2 V
- + Έχει αντοχή στην κρούση και στην ώθηση



[Εικ.24 – Σερβομηχανισμός με παρελκόμενα]

Είναι ένας από τους πολλούς που μπορεί να βρει κάποιος στο εμπόριο, σε απολύτως προσιτές τιμές και με πολύ παρόμοια χαρακτηριστικά. Η επιλογή του συγκεκριμένου ήταν τυχαία, καθόσον δεν διαθέτει κάποια εξεζητημένα χαρακτηριστικά, το μόνο το οποίο εξετάστηκε πριν την προμήθειά του ήταν το εύρος της κίνησής του, το οποίο θέλαμε να είναι από 0 έως 180° και όχι 0 – 360°, διότι η κίνηση που θα χρειαστούμε για τη λειτουργία του TheRaven είναι της τάξης περίπου των 80°.

## 5. Raspberry Pi case

Η θήκη [Εικ.25] στην οποία προστατεύεται το RPi μας είναι κατασκευασμένη από σκληρό πλαστικό, παρέχοντας επαρκή προστασία και επιτρέποντας την πλήρη πρόσβαση στον υπολογιστή.



[Εικ.25]

Είναι και αυτή της RaspberryPi Foundation, με αποτέλεσμα να εξασφαλίζει απόλυτη εφαρμογή και προστασία του υπολογιστή επιτρέποντας στο χρήστη να τον χρησιμοποιεί ενώ είναι μέσα στη θήκη του.

## 6. Λοιπά υλικά DIY κατασκευής

Τα υλικά και τα εργαλεία που χρησιμοποιήθηκαν για την υπόλοιπη κατασκευή είναι κοινά που οποιοσδήποτε μπορεί να τα προμηθευτεί από την αγορά, και έχουν όπως παρακάτω:

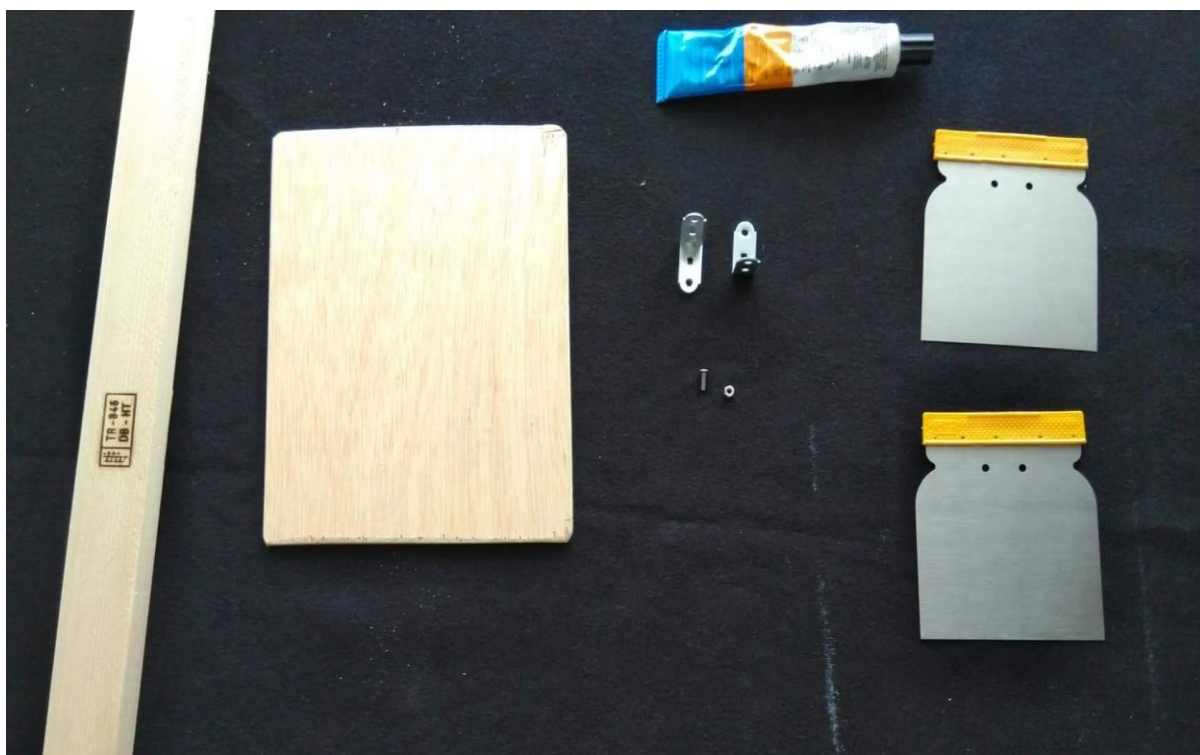


[Εικ.26]

Όπως φαίνεται παραπάνω [Εικ.26], τα εργαλεία που κυρίως χρησιμοποιήθηκαν για την κατασκευή του TheRaven είναι σέγα, πένσα, κολλητήρι, πιστόλι θερμικής κόλλας, μέτρο – γωνία, στρεβλωμένο μυτοτσιμπηδο, γαλλικός κάβουρας και κατσαβίδια διαφόρων τύπων και μεγεθών.



Παρακάτω, είναι οι πρώτες ύλες [Εικ.27] που χρησιμοποιήσαμε για να κατασκευάσουμε το TheRaven, και είναι ξυλεία, σπάτουλες χειρός, μεταλλικές γωνίες και βενζινόκολλα.



[Εικ.27]

Έτσι, εφόσον συγκεντρώθηκαν όλα τα παραπάνω, ξεκίνησε η κατασκευή του TheRaven, με απλούστατη μεν τεχνοτροπία, αλλά με ένα ικανοποιητικό αποτέλεσμα, ούτως

ώστε να διαφαίνεται ο τρόπος λειτουργίας του TheRaven, καθώς επίσης και ο σκοπός τον οποίο επιτελεί.






./.

## ΚΕΦΑΛΑΙΟ 4

### Βήματα της Κατασκευής

#### 1. Εισαγωγή

Το επόμενο βήμα είναι να κατασκευαστεί ο φορέας, πάνω στον οποίο θα ενταχθεί το RPi, η PiCamera, το Servo αλλά και η τοποθέτηση δύο πτερυγών οι οποίες θα μετακινούνται ταυτόχρονα και παράλληλα, ανάλογα με την κίνηση του servo. Τα υλικά που χρησιμοποιήθηκαν είναι απλά υλικά που μπορεί να βρει κάποιος εύκολα στο εμπόριο και έχουν όπως παρακάτω:

-  Ένα κομμάτι κόντρα πλακέ 2mm διαστάσεων 1,20x0,20 m.
-  Δύο (2) σπάτουλες χειρός.
-  Βίδες – παξιμάδια.
-  Hot glue.
-  Μεντεσέδες και γωνίες.

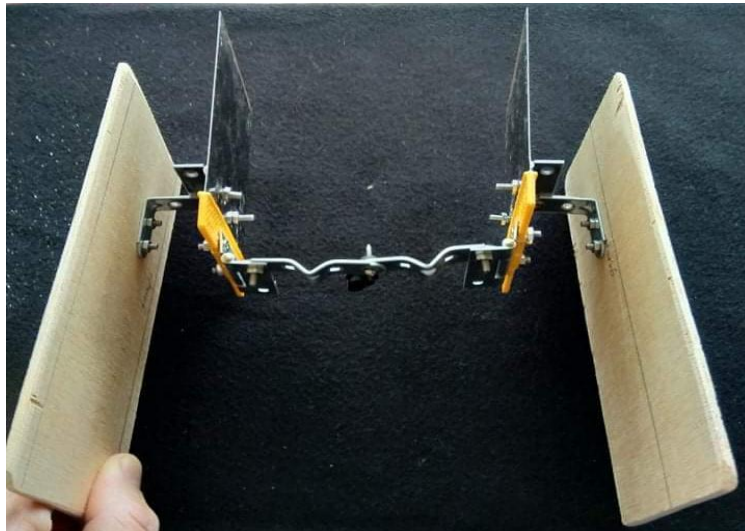
Έτσι, η κατασκευή μας, απεικονίζει ένα οπίσθιο τμήμα ενός βλήματος – πυραύλου μεγάλου βεληνεκούς, ο οποίος κατευθύνεται από το σύστημα μας.

#### 2. Κατασκευή τους Κυρίως Μέρους του TheRaven

Σε πρώτη φάση, ενώσαμε τις δύο πτέρυγες, ούτως ώστε να ακολουθούν παράλληλη κίνηση και να κινούνται ταυτόχρονα από έναν και μόνο σερβομηχανισμό. Αυτό το πετύχαμε ισιώνοντας δύο γωνίες και τις συνδέσαμε με μία βίδα και την κεφαλή του σερβομηχανισμού, τα δύο άκρα ενώθηκαν με δύο μεντεσέδες οι οποίοι με τη σειρά τους πακτώθηκαν πάνω στις πτέρυγες. Έπειτα, με τη χρήση πάλι γωνιών και μεντεσέδων, στερεώθηκαν οι πτέρυγες πάνω σε δύο επιφάνειες ξύλου οι οποίες θα αποτελέσουν το πλαίσιο στήριξης της κατασκευής μας.

Επομένως, κατασκευάστηκε το κεντρικό κομμάτι του TheRaven και το οποίο φαίνεται παρακάτω [Εικ. 28].





Ουσιαστικά, το παραπάνω τμήμα της κατασκευής θα είναι αυτό το οποίο θα μεταφράζει την λειτουργία και τις εντολές του RaspberryPi, μέσω του σερβομηχανισμού, ο οποίος θα κινεί τις πτέρυγες της κατασκευής. Πρόκειται για μία κατασκευή με απλά υλικά και δομή, η οποία όμως θα υποστηρίξει επαρκώς το ανατεθέν έργο.

### 3. Κατασκευή του Φορέα – Βάσης Στήριξης

Σειρά είχε η ολοκλήρωση της βάσης στήριξης του TheRaven, ενώνοντας τις δύο ξύλινες πλάκες στήριξης των πτερύγων, πάλι με τη χρήση ξυλείας, πιο στιβαρής αυτή τη φορά. Αφού έγιναν οι απαραίτητες μετρήσεις, κόψαμε τα απαραίτητα κομμάτια ούτως ώστε να στηρίξουν επαρκώς την κατασκευή μας και να μπορεί να λειτουργήσει χωρίς κανένα πρόβλημα.

Αφού έχουμε τα απαραίτητα μέρη, δεν μας μένει παρά να κατασκευάσουμε τη βάση μας με τη χρήση των δύο ξύλων. Θα χρησιμοποιήσουμε ξυλόβιδες για την άρτια σταθεροποίηση της κατασκευής μας και θα συγκολλήσουμε με βενζινόκολλα τα διαφορετικά μέρη για την κάλυψη όλων των κενών και την επαύξηση της ‘ομοιογένειας’ της βάσης.

Επομένως, με τα παραπάνω βήματα, φτάσαμε στην κατασκευή του TheRaven, ούτως ώστε να μπορούμε να το συνδέσουμε με το σερβομηχανισμό μας και με τη λειτουργία του TheRaven να κινούνται οι πτέρυγες και να έχουμε κατεύθυνση του πυραύλου μας.

Αυτό που συνειδητοποιήσαμε φτάνοντας σε αυτό το φαινομενικά τελικό στάδιο, είναι ότι δεν έγινε σωστός υπολογισμός στο ύψος που πακτώθηκαν οι πτέρυγες, με αποτέλεσμα να μην ταιριάζει με το ύψος του σερβομηχανισμού μας. Έτσι, μετά από κάποιες τροποποιήσεις, επιδιορθώθηκε το παραπάνω, αφού ‘κατέβηκε’ η θέση των πτερύγων και αφαιρέθηκε το περισσευούμενο πλάτος της βάσης στήριξης της κατασκευής μας. Τέλος, η βαφή της κατασκευής [Εικ.29] προσδίδει μία ομοιογένεια υλικών κατασκευής και μία ολότητα κατασκευής.

Έτσι, με λίγα και απλά υλικά έχουμε μία κατασκευή η οποία, προφανώς δεν δύναται να έχει ουσιαστική χρησιμοποίηση, αλλά είναι ικανή να δείξει ποια είναι ακριβώς η λειτουργία του TheRaven. Πώς δηλαδή βλέπει η κάμερα, μεταδίδει τα αποτελέσματα στον επεξεργαστή, αυτός με τη σειρά του κρίνει ότι ο στόχος εντοπίστηκε, προσδιορίζει τη θέση του στόχου σε σχέση με την πορεία του βλήματος. Αυτό που πραγματικά φαίνεται με την προσάρτηση του RaspberryPi πάνω στην κατασκευή μας, είναι η κατεύθυνση των πτερύγων και η αλλαγή της πορείας του βλήματος – πυραύλου, με σκοπό να προσικρούσει στον στόχο



που έχει εντοπίσει το TheRaven.

./.

## ΚΕΦΑΛΑΙΟ 5

### Software Analysis

#### 1. Εισαγωγή

Το λογισμικό το οποίο εγκαταστάθηκε στο Raspberry Pi και τα προγράμματα τα οποία χρησιμοποιήθηκαν για να λειτουργήσει το TheRaven, είναι ανοιχτού κώδικα, μπορεί να τα βρει και να τα χρησιμοποιήσει οποιοσδήποτε. Ο στόχος δεν ήταν να χρησιμοποιηθεί κάτι εξεζητημένο, αλλά αντίθετα, να αποδειχθεί πως με απλά και συνήθη ‘εργαλεία’, στα οποία μπορεί να έχει πρόσβαση οποιοσδήποτε χρήστης του Διαδικτύου, γίνεται να δημιουργήσει πολλά και διαφορετικά πράγματα – κατασκευές – εργαλεία, σε οποιοδήποτε τομέα της τεχνολογίας επιθυμεί. Έτσι, παρακάτω φαίνεται πώς ακριβώς έγινε το ‘στήσιμο’ του υπολογιστή μας, τι προγράμματα εγκαταστάθηκαν, από πού έγινε η προμήθεια αυτών, αλλά και πώς ακριβώς έγινε η εργασία για την επίτευξη της συγγραφής του κώδικα. Εν ολίγοις, όλες οι λεπτομέρειες και τα βήματα της ηλεκτρονικής κατασκευής του TheRaven φαίνονται παρακάτω.

#### 2. Λειτουργικό Σύστημα (OS)

Το λειτουργικό σύστημα είναι το προγραμματιστικό και γραφικό περιβάλλον μέσα στο οποίο ο ηλεκτρονικός υπολογιστής εκτελεί όλες τις διεργασίες του. Συμπεριλαμβανομένου τόσο των διεργασιών που ‘ζητά’ ο χρήστης απευθείας, όσο και οι εργασίες που εκτελεί ο υπολογιστής στο ‘παρασκήνιο’, εκτελούνται με χρησιμοποιώντας τα εργαλεία και τις δομές του εκάστοτε υφισταμένου Λειτουργικού Συστήματος.

Επομένως, η σημαντικότητα της ορθής επιλογής Λειτουργικού Συστήματος από το χρήστη είναι τεράστια και πρέπει να γίνεται με γνώμονα το είδος και το σκοπό χρήσης του ηλεκτρονικού υπολογιστή. Εάν π.χ. κάποιος θέλει τον υπολογιστή του για απλή επαγγελματική χρήση και τίποτα περεταίρω, θα μπορούσε να χρησιμοποιεί μία οποιαδήποτε

Linux έκδοση λειτουργικού συστήματος, καθόσον θα εκτελούσε όλες τις εργασίες που επιθυμούσε χωρίς κανένα πρόβλημα και δεν θα είχε το πρόβλημα να προσβληθεί από οποιοδήποτε ιό, γεγονός που σε επαγγελματικό περιβάλλον συμβαίνει συχνά λόγω μεταφοράς δεδομένων μέσω usb. Αντίστοιχα, εάν ο χρήστης είναι οπαδός των παιχνιδιών σε υπολογιστή και χρησιμοποιεί τον υπολογιστή του για Gaming σκοπούς, τότε θα ήταν φρόνιμο να χρησιμοποιεί περιβάλλον Windows και προτιμότερον θα ήταν να χρησιμοποιεί κάποια από τις τελευταίες εκδόσεις των Windows. Έτσι, θα είχε την απαιτούμενη απόδοση, που είναι σχετικά υψηλή όταν πρόκειται για Gaming, καθόσον θα μπορούσε να 'τρέχει' όλες τις απαραίτητες διεργασίες ταυτόχρονα και χωρίς να τις έχει 'απαιτήσει' ο ίδιο ο χρήστης από τον ηλεκτρονικό υπολογιστή. Από την άλλη, εάν ένας χρήστης επιθυμεί να κάνει καθημερινή μέση χρήση διαδικτύου και κάποιας βασικής εργασίας στον ηλεκτρονικό του υπολογιστή, τότε η χρήση του iOS θα ήταν μία πολύ καλή επιλογή. Το iOS είναι ένα λειτουργικό σύστημα που χρησιμοποιεί η Apple σε όλες τις ηλεκτρονικές της συσκευές (smartphone, tablet, laptop, desktop) και πρόκειται ουσιαστικά, για μία έκδοση Linux προσαρμοσμένη στην έλλειψη βαθιάς γνώσης προγραμματισμού του μέσου χρήστη, με πολύ καλά αποτελέσματα απόδοσης, προστασίας από Ιούς και ευχρηστίας σε διάφορες χρήσεις της σύγχρονης ζωής. Το μοναδικό σκούρο σημείο του iOS είναι η συνδεσιμότητα με οποιοδήποτε άλλο λογισμικό, παρὰ μόνο ότι εκδίδεται από την ομώνυμη εταιρεία, με αποτέλεσμα ένα μεγάλο ποσοστό χρηστών να αποφεύγει τη χρήση του γι' αυτό και μόνο το λόγο. Ένα τελευταίο παράδειγμα θα μπορούσε να είναι ένας server ο οποίος λειτουργεί 24/7, με σκοπό να υποστηρίζει συνεχώς μία ή πολλές λειτουργίες που αφορούν παραγωγή, εργασίες διαδικτύου, κτλ. Σε αυτήν την περίπτωση, θα χρησιμοποιούνταν μία έκδοση Linux, διότι οι ανάγκες για επανεκκίνηση και για προσαρμογή ρυθμίσεων λειτουργίας του υπολογιστή είναι ελάχιστες σε σχέση με οποιοδήποτε άλλο λειτουργικό σύστημα.

Το λειτουργικό σύστημα που χρησιμοποιήθηκε είναι το Raspbian OS, το οποίο είναι μία έκδοση Linux από την Debian, αποκλειστικά για το Raspberry Pi. Αποτελεί την επίσημη πρόταση της RaspberryPi Foundation καθόσον, μεγιστοποιείται η απόδοση του εν λόγω ηλεκτρονικού υπολογιστή και επιμηκύνεται η διάρκεια της φυσιολογικής ζωής του

υπολογιστή. Το βρίσκει κάποιος στο site της Raspberry και είναι open source, το κατεβάζει και το αποθηκεύει στην sd κάρτα που χρησιμοποιείται, καθόσον το RPi δεν διαθέτει εσωτερική μνήμη. Έτσι, ενεργοποιώντας πρώτη φορά το RPi κάνει όλες τις απαραίτητες εγκαταστάσεις και ρυθμίσεις, όπως σύνδεση στο οικείο wi-fi, ssh και vnc, ούτως ώστε να μπορεί να συνεχίσει εάν επιθυμεί να εκτελεί τις διεργασίες στο RaspberryPi εξ' αποστάσεως. Προτείνεται ο χειρισμός του εξ' αποστάσεως, αφού ο χρήστης θα κερδίσει σε ταχύτητα και ευκολία. Αυτό συμβαίνει διότι δεν αναπτύσσεται γραφικό περιβάλλον στο RaspberryPi και έτσι οι μοναδικές λειτουργίες με τις οποίες ασχολείται ο επεξεργαστής του Pi είναι αυτές και μόνο που ορίζει ο χρήστης εξ' αποστάσεως. Βέβαια, υπάρχει η δυνατότητα εγκατάστασης οποιουδήποτε άλλου λειτουργικού συστήματος, με αμφίβολη όμως ταχύτητα και ανταπόκριση από το RaspberryPi, καθώς πρόκειται για έναν υπολογιστή βασικών προδιαγραφών και δυνατοτήτων. Γι' αυτό το λόγο επιλέξαμε το Raspbian, το οποίο είναι όπως προαναφέρθηκε, μία Linux έκδοση για την περίπτωσή μας και αποδεικνύεται επαριέστατο. Είναι ένα λειτουργικό σύστημα προσαρμοσμένο τόσο στις δυνατότητες – προδιαγραφές του RaspberryPi, όσο και με πολλές εξατομικευμένες εφαρμογές εργασίας και διασκέδασης, καθιστώντας το ένα πλήρες working station, που οποιοσδήποτε χρήστης μπορεί να το χρησιμοποιεί καθημερινά για οποιοδήποτε σκοπό επιθυμεί. Όσον αφορά το γραφικό του περιβάλλον είναι παραπάνω από ικανοποιητικό, πληρώντας τις προϋποθέσεις που θέτει ο μέσος καθημερινός χρήστης για την εμφάνιση και εκτέλεση διεργασιών του υπολογιστή του. Αξίζει να σημειωθεί ότι στο παρελθόν, ο βασικός τομέας στον οποίο 'έχανε' ένα Linux λειτουργικό σύστημα σε σχέση με τα δημοφιλή Windows ήταν η εμφάνιση και η ομορφιά του γραφικού τους περιβάλλοντος. Έτσι, το Raspbian έχει κερδίσει σαφές έδαφος σε σχέση με τα Windows, όπως και όλα τα λοιπά Linux λειτουργικά συστήματα.

Από την άλλη, όσον αφορά ταχύτητα διεργασιών και ανταπόκριση σε οποιοδήποτε απαίτηση του χρήστη, με τον συγκεκριμένο επεξεργαστή πάντα, το Raspbian ανταποκρίνεται παραπάνω από ικανοποιητικά, έχοντας σας αποτέλεσμα να το καθιστά το λειτουργικό σύστημα που ταιριάζει 'γάντι' στον επεξεργαστή και τα δεδομένα του RaspberryPi Model B+. Έτσι, μπορεί ο χρήστης να εκμεταλλευτεί το RaspberryPi με

όποιον τρόπο επιθυμεί και για όποιον σκοπό απαιτείται. Γι' αυτό έγινε εγκατάσταση του Raspbian στον υπολογιστή του TheRaven, με σκοπό να γίνει η συγγραφή του κώδικα και να έχουμε την βέλτιστη δυνατή απόδοση του υπολογιστή μας κατά τη λειτουργία του TheRaven.

### 3. Γλώσσες Προγραμματισμού

Η γλώσσα προγραμματισμού είναι ο τρόπος μέσω του οποίου ο χρήστης καθορίζει, προγραμματίζει, τι ακριβώς διεργασίες θα εκτελέσει ο υπολογιστής ή τμήματα – περιφερειακά αυτού. Στο βάθος των ετών έχουν αναπτυχθεί πολλές διαφορετικές γλώσσες προγραμματισμού και όσο αναπτύσσεται ο τομέας της τεχνολογίας και των υπολογιστών, τόσο θα δημιουργούνται όλο και περισσότερες γλώσσες προγραμματισμού. Στις μέρες μας έχει ξεκινήσει ο τομέας αυτός του προγραμματισμού να εξελίσσεται και να εξειδικεύεται, με αποτέλεσμα οι γλώσσες προγραμματισμού που προκύπτουν να έχουν πιο συγκεκριμένο και στοχευμένο εύρος δυνατοτήτων. Ακολουθεί μία περιληπτική παρουσίαση των επικρατέστερων που χρησιμοποιούνται σήμερα, καθόσον είναι αδύνατο να τις συγκεντρώσουμε όλες αφού πλησιάζουν τις 300 (!) στον αριθμό.

#### 3.1 C

Είναι μία γλώσσα γενικής χρήσης, που δημιουργήθηκε πρωταρχικώς για συστήματα Unix. Χρησιμοποιείται στις μέρες μας, για κωδικοποίηση συστημάτων που χρησιμοποιούνε Unix, σε συστήματα cross – platform, σε κωδικοποίηση παιχνιδιών και λοιπών ειδικών εφαρμογών. Προτιμάται σε σχέση με την C++, αφού είναι πιο συμπαγής και 'τρέχει' πιο γρήγορα. Είναι η πιο συνηθισμένη γλώσσα προγραμματισμού, μετά την Java και αποτελεί τη βάση για τη δημιουργία πολλών άλλων, όπως η C#, Java και Javascript, PHP και Python.



### 3.2 C++

Είναι μία γλώσσα προγραμματισμού μέσου επιπέδου με το χαρακτηριστικό ότι είναι αντικειμενοστρεφής. Βασίζεται στη C, αλλά έχει πρόσθετα και βελτιώσεις που την κάνει μία πιο πολυεπίπεδη γλώσσα προγραμματισμού. Είναι η καταλληλότερη για μεγάλα έργα – εργασίες, επειδή μπορεί να διασπαστεί σε επιμέρους τμήματα που θα έχουν εύκολη συνεργασία. Χρησιμοποιείται από κολοσσούς τεχνολογίας ανά την υφήλιο, όπως Google, Adobe, Mozilla και Microsoft.

### 3.3 Objective C

Όπως και οι περισσότερες γλώσσες προγραμματισμού, έτσι και η Objective C προέρχεται από την Mother-Of-All την C. Είναι ένας κώδικας γενικής χρήσης και υψηλού επιπέδου, που διαθέτει μία πρόσθετη λειτουργία κατά την οποία ανταλλάσει μηνύματα. Είναι η γλώσσα προγραμματισμού που χρησιμοποιείται από τις εφαρμογές OS X και iOS της Apple.

### 3.4 C#

Άλλη μία γλώσσα προγραμματισμού γενικής χρήσης, αντικειμενοστρεφής, η C# δημιουργήθηκε από την εταιρεία Microsoft το 2001. Παρόλο που ονομαστικά δίνει την εντύπωση ότι ανήκει στην οικογένεια των γλωσσών προγραμματισμού C, στην ουσία όμως έχει πολλά περισσότερα κοινά με την Java παρά με οποιαδήποτε άλλη γλώσσα προγραμματισμού. Η C# χρησιμοποιείται κατά κόρον για εσωτερικές και επιχειρηματικές λύσεις και έτσι έχει γίνει αρκετά δημοφιλής, αποτελώντας τον αντίπαλο της Java.

### 3.5 Java

Η Java είναι η πιο δημοφιλής και ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού παγκοσμίως. Αρχικά, δημιουργήθηκε για να χρησιμοποιηθεί στη διαδραστική τηλεόραση, εντούτοις, κατέληξε να είναι η πιο γνωστή γλώσσα στις συσκευές Android και γενικά στις εφαρμογές του Διαδικτύου. Έχει, επίσης, χρησιμοποιηθεί επανειλημμένως σε λογισμικό επιχειρήσεων. Είναι πολλαπλών χρήσεων που μπορεί να χρησιμοποιηθεί και σε cross – platform, δηλαδή είναι το ίδιο εύκολο να χρησιμοποιηθεί σε εφαρμογές smartphone, όσο και σε αντίστοιχες desktop. Μοιάζει με την C++ στη δομή και τη σύνταξη, κάνοντας έτσι εύκολη την εκμάθησή της για κάποιον που είναι εξοικειωμένος με τις γλώσσες προγραμματισμού της ‘οικογένειας’ C.

### 3.6 Javascript

Η εν λόγω γλώσσα προγραμματισμού ως ένας κώδικας με πολλά και διαφορετικά πρόσθετα για να επεκτείνει τη λειτουργικότητα των ιστοσελίδων. Προσθέτει διαδραστικά χαρακτηριστικά σε αυτές, όπως φόρμες υποβολής, κινούμενα σχέδια, ερωτηματολόγια, κτλ. Χρησιμοποιείται κυρίως για λύσεις προγραμματισμού που αλληλεπιδρά με τον πελάτη και για front – end development. Είναι συμβατή με όλους τους browsers, καθιστώντας μία καλή επιλογή για ανάπτυξη λογισμικού ιστοσελίδων, παρά τη φήμη της για τη δυσκολία του debugging.

### 3.7 Swift

Είναι η γλώσσα προγραμματισμού που αντικατέστησε την Objective-C στην κατασκευή των προγραμμάτων της Apple. Τα τελευταία χρόνια έχει κερδίσει έδαφος διότι είναι εύκολη στο να διαβαστεί, εύκολη στο να συντηρηθεί και πιο γρήγορη στην εκτέλεση από την Objective-C. Εάν κάποιος θέλει να γράψει προγράμματα/ εφαρμογές για iOS ή να



γίνει προγραμματιστής της Apple, η Swift είναι η γλώσσα προγραμματισμού που πρέπει να γνωρίζει. Παρά το γεγονός ότι η Objective – C συνεχίζει να χρησιμοποιείται, η Swift είναι η γλώσσα που επιλέγεται από όλους σχεδόν τους προγραμματιστές της Apple.

### 3.8 PHP

Η PHP είναι ένας open source κώδικας, ο οποίος χρησιμοποιείται κατά κύριο λόγο στην δημιουργία ιστοσελίδων και γενικότερα web development. Δημιουργήθηκε για τη βελτιστοποίηση κατασκευής ιστοσελίδων, είναι σχετικά απλή γλώσσα προγραμματισμού και εύκολη στην εκμάθηση. Χρησιμοποιείται ευρέως από εταιρείες που ασχολούνται κατά βάση με το διαδίκτυο, εταιρείες όπως η Facebook, WordPress και Wikipedia.

### 3.9 Ruby

Η Ruby είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού, γενικής χρήσεως που αναπτύχθηκε στην Ιαπωνία κατά τα μέσα της δεκαετίας του '90. Είναι μία από τις απλούστερες γλώσσες προγραμματισμού και αποτελεί μία πολύ καλή επιλογή για να ξεκινήσει κάποιος τον προγραμματισμό. Αναπτύχθηκε για να αυξήσει την παραγωγικότητα και με επιδίωξη να είναι ευχάριστη στην κωδικοποίηση. Η εκμάθησή της συνίσταται ως η πρώτη γλώσσα προγραμματισμού για κάποιον αρχάριο, και γι' αυτό είναι ευρέως γνωστή. Χρησιμοποιείται από δημοφιλή sites όπως τα Shopify, Airbnb και Hulu.

### 3.10 Python

Ένας άλλος υψηλού επιπέδου, γενικής χρήσης κώδικας, η Python, είναι από τις προτιμότερες των ανερχόμενων προγραμματιστών. Είναι ευχάριστη στη χρήση και αποτελεί άλλη μία scripting γλώσσα προγραμματισμού που συνιστάται ανεπιφύλακτα σε αρχάριους προγραμματιστές, αποτελώντας την κορυφαία εισαγωγική γλώσσα προγραμματισμού στα

αμερικανικά πανεπιστημιακά προγράμματα. Η χρήση της είναι κυρίως σε εφαρμογές του διαδικτύου και για την ασφάλεια πληροφοριακών συστημάτων και έχει γίνει ιδιαίτερα δημοφιλής στην ακαδημαϊκή κοινότητα για την ανάλυση δεδομένων. Δεν είναι τυχαίο άλλωστε, που κολοσσοί του διαδικτύου όπως η Google, Dropbox και Spotify, χρησιμοποιούν την Python.

#### 4. Επιλογή Κατάλληλης Γλώσσας Προγραμματισμού

Η επιλογή της κατάλληλης γλώσσας προγραμματισμού για την κατασκευή και τη λειτουργία του TheRaven, καθόσον είναι αυτή που θα καθορίσει τι δυνατότητες θα έχουμε να εκτελέσουμε, αλλά και πόσο εύκολα θα το πετύχουμε αυτό. Έτσι, από την τεράστια γκάμα που διατίθενται θα γίνει επιλογή της πιο εύχρηστης και προσιτής γλώσσας, η οποία εξασφαλίζει τα επιθυμητά αποτελέσματα.

Η ‘οικογένεια’ των γλωσσών προγραμματισμού που ανήκουν στην C, δηλαδή οι C, C++, C#, Objective C, είναι η βάση όλων των υπολοίπων και θα ήταν μία επιλογή σίγουρη και αποτελεσματική. Αυτό που όμως αποτελεί μειονέκτημά τους είναι το μέγεθος, καθόσον είναι πολύ πιο μακροσκελείς από άλλες γλώσσες, με αποτέλεσμα να απαιτείται ο διπλάσιος ή και παραπάνω αριθμός γραμμών κώδικα – εντολών για να εκτελέσουν κάποιες διεργασίες. Επομένως, θα δημιουργούσε πρόβλημα χρόνου και περισσότερης εργασίας σε σχέση με άλλες πιο εξελιγμένες γλώσσες προγραμματισμού.

Αντίστοιχα, οι γλώσσες προγραμματισμού Java και Javascript, είναι γλώσσες προγραμματισμού που εξειδικεύονται στη διαδραστική σχέση ιστοσελίδας και χρήστη, δίνοντας την δυνατότητα στο χρήστη να κατευθύνει την εργασία του στα ενδιάμεσα βήματά της και να καταλήγει εκεί που επιθυμεί, έχοντας πάντα τη δυνατότητα να επιστρέψει σε προηγούμενα βήματα και να μετατρέψει την πορεία του. Σαν ιδέα λειτουργίας, θα ήταν χρήσιμο σε περίπτωση που το TheRaven λειτουργούσε με κάποιο τρόπο τηλεχειρισμού και δυναμικής απόφασης – επιλογής του ανθρώπου – χειριστή. Αντιθέτως, η λειτουργία του TheRaven προσφέρει χαρακτήρα αυτονομίας και επεξεργασίας των δεδομένων από το ίδιο,

χωρίς ουδεμία επέμβαση του χειριστή από τη στιγμή που θα ενεργοποιηθεί μέχρι τέλους... Έτσι, οι παραπάνω γλώσσες προγραμματισμού είναι ακατάλληλες για τον προγραμματισμό του TheRaven.

Μία άλλη υφιστάμενη γλώσσα προγραμματισμού που προαναφέρθηκε, είναι η Swift, η οποία χρησιμοποιείται κατά κόρον από την εταιρεία Apple. Έχει εξειδικευτεί στο λογισμικό iOS και τις εφαρμογές που χρησιμοποιεί, με αποτέλεσμα να καθιστά απαραίτητη την εξεζητημένη βαθιά γνώση της για τη δημιουργία κάποιων εντολών που δεν διατίθενται στις εμφανείς επιλογές της. Επομένως, θα οι απαιτήσεις του προγραμματισμού του TheRaven σε Swift θα ήταν σχετικά υψηλές, με τον κίνδυνο να καταστεί ακόμη και αδύνατο το εν λόγω εγχείρημα.

Η γλώσσα προγραμματισμού PHP από την άλλη, χρησιμοποιείται κατά βάση για την κατασκευή ιστοσελίδων, γεγονός που προβάλλει πολλά εμπόδια στον προγραμματισμό εντολών και διεργασιών που απαιτεί το TheRaven. Αυτό συμβαίνει διότι η δομές που έχουν να κάνουν με γραφιστικό περιβάλλον και αλληλεπίδρασης χρήστη – υπολογιστή δεν ικανοποιούν τις απαιτήσεις του TheRaven. Επομένως, προκύπτει η ανάγκη απόρριψης και αυτής της γλώσσας προγραμματισμού..

Άλλη μία πιθανή επιλογή για τον προγραμματισμό του TheRaven είναι η Ruby. Όπως προαναφέρθηκε, δημιουργήθηκε στην Ιαπωνία, γεγονός που καθιστά τις λειτουργίες της διαφορετικές σε σχέση με τις πιο δημοφιλείς γλώσσες προγραμματισμού, όπως επίσης, πρέπει να αναφερθεί και το γεγονός ότι η εύρεση γνώσεων και πακέτων κώδικα για την εν λόγω γλώσσα είναι σχετικά περιορισμένη. Συμπερασματικά, δεν θα επιλέγαμε μία γλώσσα προγραμματισμού που είναι σχετικά out-of-date, ως προς τις δυνατότητες εξέλιξης.

Συμπερασματικά, καταλήγουμε στην γλώσσα προγραμματισμού που ακούει στο όνομα Python. Πρόκειται για μία γλώσσα που ικανοποιεί τις απαιτήσεις του προγραμματισμού του TheRaven, είναι πιο απλή και ευέλικτη από όλες τις γλώσσες προγραμματισμού της ‘οικογένειας’ C και είναι καθιστά εφικτή τη χρήση της από έναν σχετικά νέο προγραμματιστή. Είναι ανερχόμενη, χαρακτηριστικό το οποίο χαρακτηρίζει όλο το εγχείρημα του TheRaven, όσον αφορά το Hardware, την αντίληψη της δημιουργίας,

το σκοπό αυτής και το τρόπο λειτουργίας. Μπορεί κάποιος να βρει πληθώρα δεδομένων και πακέτων κώδικα της Python στο διαδίκτυο, δίνοντας έτσι τη δυνατότητα στον χρήστη να ‘δημιουργήσει’ και να εκτελέσει μία τεράστια γκάμα από λειτουργίες και εντολές. Τέλος, η Python είναι μία γλώσσα προγραμματισμού που διακρίνεται από την προσβασιμότητα και τη δυνατότητα τροποποιήσεων σε όλα τα επιμέρους τμήματά της, πράγμα που προσδίδει ευχέρεια στο εγχείρημά μας, καθόσον από τις δοκιμές πιθανόν να προκύψει η παραπάνω ανάγκη.

### 5. Βιβλιοθήκες Νευρωνικών Δικτύων

Το περιβάλλον μέσα στο οποίο το TheRaven θα εκτελεί όλες τις διεργασίες του πρέπει να είναι τέτοιο ώστε να εκμεταλλεύεται την εικόνα ή το video και έπειτα να κάνει τη στοχοποίηση και όλες τις περεταίρω λειτουργίες που επιθυμούμε. Το περιβάλλον που μπορεί να προσφέρει αυτή τη δυνατότητα έχει να κάνει με το Computer Vision, με την ικανότητα δηλαδή της τεχνολογίας να ‘βλέπει’ και από εκεί και πέρα να έχει την δυνατότητα να κρίνει και να αξιολογεί αυτό που βλέπει. Γίνεται λόγος για περιβάλλοντα που ανοίγουν έναν δρόμο για ατελείωτες νέες δυνατότητες και λειτουργίες που πλέον, ένας κατάλληλα προγραμματισμένος υπολογιστής, σε συνεργασία με μία κάμερα υψηλής ανάλυσης, δύναται να εκτελέσει εργασίες αντικαθιστώντας το ανθρώπινο μάτι. Μάλιστα, εάν ληφθεί υπόψη και το γεγονός ότι ο υπολογιστής μπορεί να εκτελεί μεγάλο αριθμό εργασιών ταυτόχρονα και χωρίς να κουράζεται, τότε μπορεί να εκτελεί την εργασία καλύτερα από τον άνθρωπο. Ένα παράδειγμα των παραπάνω είναι η αναγνώριση προσώπου από ένα κλειστό κύκλωμα παρακολούθησης π.χ. σε ένα αεροδρόμιο, ψάχνοντας κάποιο συγκεκριμένο πρόσωπο του οποίου βέβαια τα βιομετρικά στοιχεία έχει ο υπολογιστής στη μνήμη του. Έτσι, πάνω σ’ αυτή την τεχνολογική εξέλιξη της πληροφορικής θα εργάζεται το TheRaven για να εκτελεί τον εντοπισμό του στόχου που επιθυμεί ο προγραμματιστής και να μετατρέπει τον εντοπισμό σε κίνηση του βλήματος – πυραύλου, με σκοπό να πετύχει αυτό το στόχο του.

### 5.1 Open Source Computer Vision Library (OpenCV)

Το λογισμικό OpenCV βοηθά το χρήστη να εκμεταλλεύεται την λήψη video και να εκτελεί ποικίλες εργασίες μέσω αυτού, όπως να εντοπίζει συγκεκριμένα πράγματα, να ανιχνεύει κίνηση, να εντοπίζει ένα συγκεκριμένο χρώμα και γενικώς πλήθος διεργασιών. Ουσιαστικά γίνεται λόγος για ένα λογισμικό ανοικτής πηγής, που οποιοσδήποτε μπορεί να έχει πρόσβαση, και το οποίο είναι μία βιβλιοθήκη λογισμικού που έχει να κάνει με το computer vision και το machine learning. Κατασκευάστηκε για να διευκολύνει την επικοινωνία μεταξύ υπολογιστή – ανθρώπου – λοιπών μηχανημάτων τεχνολογίας, καθώς επίσης και να επιταχύνει τις διεργασίες των μηχανών στις γραμμές παραγωγής. Επίσης, ένα σημείο που ο κατασκευαστής έδωσε βαρύτητα ήταν η ευκολία στην τροποποίηση του κώδικα, ούτως ώστε, ο ενδιαφερόμενος προγραμματιστής, να έχει την άμεση δυνατότητα να τροποποιήσει τον κώδικα και να πετύχει τις λειτουργίες που επιθυμεί, αναλόγως της εργασίας του. Η βιβλιοθήκη του OpenCv έχει περισσότερους από 2500 ενσωματωμένους αλγόριθμους, και καλύπτει μία τεράστια γκάμα διεργασιών που περιλαμβάνουν μεταξύ άλλων και: εντοπισμό και αναγνώριση προσώπου, ταυτοποίηση αντικειμένων, κατηγοριοποίηση βιντεοσκοπημένων ανθρωπίνων ενεργειών, ‘κλείδωμα’ κίνησης κάμερας, ‘κλείδωμα’ κινούμενου αντικειμένου, εξαγωγή 3διάστατων μοντέλων αναπαράστασης αντικειμένων, ‘ράψιμο’ διαφορετικών εικόνων για δημιουργία ενός ειδώλου υψηλής ανάλυσης ή μεγάλου μεγέθους, εξάλειψη κόκκινων ματιών από φωτογραφίες που λήφθηκαν με τη χρήση φλας, ακολουθία κίνησης ματιών και πάρα πολλές ακόμη λειτουργίες. Το εν λόγω λογισμικό απαριθμεί περίπου 14 εκατομμύρια downloads και χρησιμοποιείται ευρέως τόσο από εταιρίες κολοσσούς στους τομείς τους, όπως η Google, Yahoo, Microsoft, Intel, IBM, Honda, Toyota, αλλά και από μικρότερες αναπτυσσόμενες εταιρίες οι οποίες έχουν στηρίξει όλη τη λειτουργία τους πάνω στο OpenCv. Ουσιαστικά με αυτού του είδους το λογισμικό μπορεί πλέον ο χειριστής να εκμεταλλευτεί την εικόνα ή το βίντεο σαν να ήταν μία λίστα από διαφορετικές λειτουργίες που τόσα χρόνια τις βλέπαμε σαν ένα αναπόσπαστο πακέτο, που τώρα όμως έχουμε τη δυνατότητα να διαλέξουμε, να μετατρέψουμε, να αφαιρέσουμε, να

προσθέσουμε, να δημιουργήσουμε, να εκτελέσουμε οποιαδήποτε λειτουργία επιθυμούμε, εκμεταλλευόμενοι την εικόνα και το βίντεο. Διαθέτει λειτουργίες συμβατές με C++, Python, MATLAB και Java, και δύναται να λειτουργήσει σε Windows, Linux, Android και iOS. Το OpenCV διαθέτει διάφορες εκδόσεις, οι οποίες ουσιαστικά αναφέρονται σε βελτιώσεις που γίνονται, με απλουστευμένες διεργασίες, περισσότερα πακέτα κώδικα και καινούργιες ανάγκες που προκύπτουν από την παράλληλη εξέλιξη της τεχνολογίας. Γίνεται λόγος, εν ολίγοις για διαρκείς αναβαθμίσεις και τροποποιήσεις που ενσωματώνει στον κώδικά του, με σκοπό τη βελτίωση και την δυνατότητα να ανταπεξέρχεται στις ολοένα και περισσότερες τεχνολογικές απαιτήσεις.

## 5.2 Keras

Μία άλλη βιβλιοθήκη νευρωνικών δικτύων ανοιχτού κώδικα, που είναι διαθέσιμη και συναντάται ολοένα και περισσότερο στον κόσμο των προγραμματιστών είναι το Keras. Έχει συνταχθεί στην γλώσσα προγραμματισμού Python κατά το 2015, με αφορμή τις βιβλιοθήκες ανοιχτού κώδικα TensorFlow και CNTK. Και οι δύο έχουν να κάνουν με επεξεργασία εικόνας και ανάλυσης δεδομένων εικόνας και βίντεο. Ο δημιουργός του εξηγεί ότι το Keras κατασκευάστηκε για να αποτελεί περισσότερο μία διεπαφή (interface) παρά ένας υπολογιστής αυτόνομος που θα εφαρμόζει το machine learning και θα λειτουργεί χωρίς την ανθρώπινη παρέμβαση. Το Keras περιέχει μία μεγάλη γκάμα συμπληρωματικών νευρωνικών δικτύων και πολλά εργαλεία επεξεργασίας και εκμετάλλευσης εικόνας και κειμένου. Πρόκειται για μία βιβλιοθήκη που χρησιμοποιεί το Deep learning και υποστηρίζει όλων των ειδών τα νευρωνικά δίκτυα, με τρόπο και απαιτούμενες διαδικασίες φιλικές προς το χρήστη και σχετικά εύκολο για κάποιον αρχάριο προγραμματιστή. Αυτό το πετυχαίνει με ελαχιστοποίηση της ανθρώπινης παρέμβασης κατά τη διάρκεια των διεργασιών του, αλλά και με την παροχή έγκαιρης και ικανοποιητικής σε ποσότητα, ανάδρασης (feedback). Επίσης, η δομή των δεδομένων του Keras είναι σε ‘κομμάτια’ κώδικα (modules) τα οποία είναι ανεξάρτητα το ένα από το άλλο, και παρέχει τη δυνατότητα στο χρήστη να τα συνδέει και να



τα συνδυάζει, αναλόγως του έργου που θέλει να εκτελέσει. Όμως, η μοναδική γλώσσα προγραμματισμού που μπορεί κάποιος να χρησιμοποιήσει για το Keras είναι η Python, κάνοντας σε ορισμένες περιπτώσεις αδύνατη τη χρήση του. Βέβαια, υπάρχει η δυνατότητα σε ορισμένες περιπτώσεις να χρησιμοποιήσει κάποιος το Keras σε εφαρμογές για smartphone σε Android και iOS. Η ανάπτυξή του, βέβαια, βρίσκεται σε πρώιμο στάδιο και ανά τακτά χρονικά διαστήματα εκδίδονται βελτιωμένες εκδόσεις, όπως και στο OpenCV, με σκοπό πάντα την αύξηση της ευκολίας χρήσης και την κάλυψη περισσότερων τεχνολογικών αναγκών. Ωστόσο, απαριθμεί περισσότερους από 200.000 χρήστες από τον Νοέμβριο του 2017 και έπειτα, χρονολογία κατά την οποία, η Google αποφάσισε να συνεργαστεί και να υποστηρίξει το Keras στην βιβλιοθήκη της, TensorFlow. Αξίζει να σημειωθεί ότι η ομώνυμη εταιρεία προσφέρει στο επίσημο site της δωρεάν τρόπους εκμάθησης στην μέθοδο Deep Learning, με σκοπό την εξοικείωση του χρήστη στο περιβάλλον, τις διαδικασίες και τις μεθόδους του Keras.

#### 6. Επιλογή Κατάλληλης Βιβλιοθήκης Νευρωνικών Δικτύων

Από τα παραπάνω συμπεραίνει κάποιος ότι, το λογισμικό OpenCV είναι ότι ακριβώς χρειάζεται για να λειτουργήσει το TheRaven, ούτως ώστε να αναγνωρίζει, σε πρώτο χρόνο, το στόχο μας και έπειτα, να συνεχίζουν οι λειτουργίες που επιθυμούμε. Ο λόγος που επιλέχθηκε το OpenCV για την κατασκευή της εν λόγω εργασίας είναι, διότι είναι πιο δυσεύρετα τα δεδομένα του Keras και ακριβώς επειδή είναι σε πιο υβριδικό στάδιο, οι μέθοδοι και οι διαδικασίες είναι άγνωστες στο ευρύ κοινό με αποτέλεσμα να απαιτεί πιο εξειδικευμένη και βαθιά γνώση προγραμματισμού και λογισμικού. Επίσης, το OpenCV προσφέρει τη δυνατότητα πιο άμεσης και εύκολης επέμβασης στον κώδικα, ούτως ώστε ο χρήστης να μπορεί στο μέλλον να εμπλουτίσει ή να διαφοροποιήσει κατ' οιονδήποτε τρόπο τον κώδικα του TheRaven. Ένας άλλος λόγος που επιλέχθηκε εφαρμόστηκε το OpenCV είναι η συμβατότητά του με περισσότερες γλώσσες προγραμματισμού και διαφορετικά λογισμικά, γεγονός που επιτρέπει την πληρέστερη μελλοντική του εκμετάλλευση. Έτσι, μέσα

στο περιβάλλον του OpenCV έχουμε τη δυνατότητα να εντοπίσουμε, να αναγνωρίσουμε και να 'κλειδώσουμε' το αντικείμενο που επιθυμούμε, με ευκολία διαχείρισης και τροποποίησης του κώδικα κατά το δοκούν. Η ύπαρξη πακέτων κώδικα, για το εν λόγω περιβάλλον, διευκολύνουν αισθητά τον προγραμματισμό του TheRaven, εξοικονομώντας χρόνο από τη δημιουργία δεδομένων που υπάρχουν, ενώ στην περίπτωση του Keras θα έπρεπε να γίνει 'ράψιμο' διαφορετικών modules με αμφίβολα αποτελέσματα και περιττές προσπάθειες που πολλές φορές πιθανόν να κατέληγαν σε αποτυχία..

## 7. Ολοκληρωμένο Λογισμικό

Συνοψίζοντας, το πακέτο των απαιτούμενων προγραμμάτων, εφαρμογών, λογισμικού γενικότερα, συγκεντρώθηκε με γνώμονα την συμβατότητα μεταξύ των και προφανώς, την αποτελεσματικότητα. Επίσης, λήφθηκε υπόψη η ευστοχία των επιμέρους διεργασιών, αποφεύγοντας περιττές εργασίες αλλά και περιττά δεδομένα αρχείων και προγραμμάτων, καθόσον, όπως προαναφέρθηκε, ο χρησιμοποιούμενος υπολογιστής είναι περιορισμένων δυνατοτήτων και ελαχίστων προδιαγραφών. Έτσι, στο περιβάλλον του Raspbian γίνεται εγκατάσταση του OpenCV και η συγγραφή του κώδικα γίνεται σε γλώσσα προγραμματισμού Python. Με αυτόν τον συνδυασμό, έχουμε την βέλτιστη βασική λειτουργία του RaspberryPi, με ένα πλήρες εκμεταλλεύσιμο ψηφιακό περιβάλλον και με το αρχείο μορφής .py δίνουμε τις επιθυμητές εντολές στο TheRaven να εκτελέσει τις απαιτούμενες επιμέρους διεργασίες. Παράλληλα, σαν επέκταση – συμπλήρωση του OpenCV, χρησιμοποιήσαμε ένα έτοιμο μοντέλο του Deep Learning, το Caffe, το οποίο είναι μία παραλλαγή που έχει κατασκευαστεί στο Πανεπιστήμιο της Καλιφόρνια, και ουσιαστικά γίνεται λόγος για μία μέθοδο του machine learning βασισμένη στην εκμάθηση της αναπαράστασης των δεδομένων, σε αντίθεση με τους πολύπλοκους αλγορίθμους που καθορίζουν συγκεκριμένες εργασίες και λειτουργίες. Το προτέρημα του Deep Learning, άρα και του Caffe, είναι ότι οι λειτουργία δύναται να διεξάγεται υπό πλήρη επίβλεψη, υπό μερική επίβλεψη, αλλά και παντελώς αυτόνομα, χωρίς επίβλεψη και οποιαδήποτε παρέμβαση του χειριστή. Επομένως,



ο επεξεργαστής που λειτουργεί με Deep Learning δύναται να είναι αυτόνομος και να εκτελεί όλες τις διεργασίες του χωρίς καμία απολύτως ανθρώπινη παρέμβαση.

Επίσης, αυτό που είναι απαραίτητο για την διεκπεραίωση της αποστολής μας είναι το `numpy` πακέτο κώδικα, είναι ένα βασικό πακέτο λειτουργιών που αφορά τον επιστημονικό λογισμό και είναι γραμμένο και συμβατό να λειτουργεί με γλώσσα προγραμματισμού Python. Ανάμεσα σε πολλές και διαφορετικές λειτουργίες περιέχει και ένα δυναμικό `array object N- διαστάσεων`, ‘έξυπνες’ λειτουργίες μετάδοσης, διάφορα εργαλεία προσαρμογής και παραλλαγής για άλλες γλώσσες προγραμματισμού όπως είναι η C++ και η C. Τέλος, περιέχει δεδομένα και λειτουργίες γραμμικής άλγεβρας, μετασχηματισμού Fourier καθώς και γεννήτρια τυχαίων αριθμών. Εκτός από την εμφανή επιστημονική χρήση του εν λόγω πακέτου, μπορεί κάποιος να χρησιμοποιήσει το NumPy σαν μία πολυδιάστατη ‘δεξαμενή’ δεδομένων κώδικα, διαφορετικού τύπου, με αποτέλεσμα να αποτελεί άλλο ένα πολυχρηστικό εργαλείο. Ένας περιορισμός με το NumPy είναι όταν έχουμε να επεκτείνουμε ένα διάνυσμα, αυτό δημιουργεί ένα άλλο διάνυσμα και δεν μετατρέπει το ήδη δοθέν. Επίσης, σε ορισμένες περιπτώσεις μπορεί να καθυστερήσει την λειτουργία του επεξεργαστή μας, διότι πρέπει ορισμένες λειτουργίες να τις μετατρέψει από σταθερές σε γραμμικές κι έτσι να έχουμε την επιβράδυνση της λειτουργίας μας.

Συνέχεια είχε το `Imutils` πακέτο το οποίο είναι μία σειρά από βολικές και χρήσιμες λειτουργίες για να εκτελεστεί επεξεργασία εικόνας, όπως ανασχηματισμό, μεγέθυνση, περιστροφή, μετάφραση, εντοπισμό ακμών, κατηγοριοποίηση περιγράμματος και πολλά ακόμη, πάντα σε συνδυασμό με OpenCV και Python.

Έτσι, εύκολα συμπεραίνει κάποιος ότι οι παραπάνω βιβλιοθήκες – πακέτα κώδικα, είναι απολύτως απαραίτητες για την λειτουργία του TheRaven, καθόσον από αυτά εξαρτάται η οπτική μετάδοση και αναγνώριση του στόχου μας.

## 8. Επιμέρους Ενέργειες

Μετά την επιλογή του κατάλληλου λογισμικού, όπως παραπάνω, σειρά είχε η υλοποίηση της ψηφιακής κατασκευής του έργου. Παρακάτω φαίνονται όλα τα βήματά της μέχρι το πέρας και την επιτυχή υλοποίηση του TheRaven.

Ενεργοποιώντας απλά το RPi, εφόσον όπως προαναφέρθηκε έχει γίνει η τοποθέτηση της sd card με αποθηκευμένο το Raspbian OS, και από τον οικιακό υπολογιστή, χρησιμοποιώντας την εφαρμογή IP Scanner, εντοπίζουμε το RPi και συνδεόμαστε μέσω SSH στο CLI του RPi. Ένας εναλλακτικός τρόπος σύνδεσης στο RPi εξ' αποστάσεως, είναι με την εφαρμογή VNC Viewer και έχουμε στην οθόνη του Laptop μας όλο το γραφικό περιβάλλον του RPi μας. Για την παρούσα εργασία χρησιμοποιήθηκε ο πρώτος τρόπος.

Έτσι, χωρίς να βρούμε το πρόβλημα του «βάρους» του γραφικού περιβάλλοντος, εγκαταστήσαμε την Python (έκδοση 2.7, κι αυτό διότι η νεότερη έκδοση 3.3.1 δεν «συνεργάζεται» άριστα με το OS Raspbian και έχει ως αποτέλεσμα τη χρονοκαθυστέρηση σε πολλές διεργασίες), γλώσσα η οποία χρησιμοποιήσαμε στην εργασία μας. Ακολούθως, έγινε η εγκατάσταση του OpenCV, ούτως ώστε να μας παρέχει το περιβάλλον που θα εκμεταλλευτούμε για τις περαιτέρω ενέργειες του TheRaven.

Στη συνέχεια, σειρά είχε η PiCamera, η οποία εγκαταστάθηκε με ευκολία και η δοκιμή της με απλά scripts, που εκτελούν φωτογράφιση ή βιντεοσκοπήση. Η κάμερα αυτή είναι κατασκευασμένη για τα δεδομένα του υπολογιστή μας, απαιτώντας ελάχιστες ποσότητες λογισμικού αλλά και τροφοδοσίας, καθόσον και οι δυνατότητες τροφοδοσίας του RaspberryPi είναι σχετικά περιορισμένες και θα απαιτούνταν εξωτερική πηγή ρεύματος σε αντίθετη περίπτωση. Βέβαια, θα μπορούσαμε να χρησιμοποιήσουμε και μία απλή webcam που διατίθεται από πολλές εταιρείες του εμπορίου, συνδέοντάς την σε μία από τις θύρες usb του RPi μας. Σ' αυτήν την περίπτωση θα απαιτούνταν τροποποιήσεις στον κώδικά μας, ούτως ώστε να διαβάζονται τα δεδομένα από τη συγκεκριμένη είσοδο του υπολογιστή μας.

Τέλος, ο σκοπός όλων των παραπάνω εργασιών ήταν να εκτελέσουμε live streaming video σε OpenCV περιβάλλον, το οποίο και έλαβε χώρα, δίνοντας σε εκείνη τη χρονική

στιγμή, τη δυνατότητα να προβάλλουμε την εικόνα από την PiCam σε ComputerVision περιβάλλον, αποδεικνύοντας ότι όλα τα παραπάνω βήματα είχαν γίνει ορθά, επιτυγχάνοντας μία φυσιολογική ροή όλων των διεργασιών και μία ομαλότητα στη λειτουργία του RaspberryPi. Ήρθε η σειρά του αρχείου που θα ‘τρέχει’ ουσιαστικά το TheRaven και θα εκτελεί όλες τις επιθυμητές διεργασίες και θα αποτελέσει τον Εντοπιστή Στόχου – Κατευθυντή Βλήματος «TheRaven».

## 9. Ανάλυση Κώδικα

Πιο αναλυτικά, η βασική ιδέα συγγραφής του κώδικα ήταν από το pyimagesearch.com και πιο συγκεκριμένα από το tutorial «A gentle guide to Deep Learning», καθώς μετά από δοκιμή διάφορων άλλων και διαπίστωση ακαταλληλότητας, κατέληξα σε αυτό καθόσον επιδέχονταν τις περισσότερες αλλαγές και ήταν πιο αναλυτικό από άλλα, δίνοντας έτσι τη δυνατότητα να παρεμβαίνει κάποιος εύκολα στον κώδικα και να τον τροποποιεί αναλόγως τις ανάγκες του. Επίσης, χρησιμοποιεί την τακτική CNN-Frame και όχι άλλη, όπως π.χ. την SSD, γεγονός που το κάνει πιο γρήγορο και ξεκάθαρο ως προς την αποτύπωση στην εικόνα – frame.

```
# import the necessary packages
from __future__ import print_function
from imutils.video import VideoStream
from imutils.video import FPS
```

Στη συνέχεια έγινε η εισαγωγή των κατάλληλων ‘πακέτων’, με πιο σημαντικά τα ‘numpy’ και ‘imutils’, καθόσον είναι ότι πιο πλήρες κυκλοφορεί στο web που αφορά στον επιστημονικό λογισμό και έχουν ενσωματωμένες λειτουργίες που είναι χρήσιμες και πολύπλοκες να γραφούν εξ’ αρχής, όπως π.χ. γραμμική άλγεβρα.

```
import numpy as np
import argparse
import imutils
import time
import cv2
import os
```

Επίσης, ακριβώς σε αυτό το σημείο πρέπει να γίνει και η εισαγωγή – ενεργοποίηση των GPIO Pins, καθόσον εάν δεν συμβεί αυτό, θα παραμείνουν ανενεργά και δεν θα αναγνωρίζεται – κινείται το servomotor μας.

```
import RPi.GPIO as GPIO
```

Έπειτα, καλούμε τα δύο μας αρχεία που συμπεριλαμβάνονται στο παραπάνω πακέτο κώδικα, τα Caffe --prototxt, που καθορίζει το είδος του αντικειμένου που εντοπίζεται, και το --model, το οποίο δίνει το ‘μονοπάτι’ για να διαβαστεί το αρχείο μας.

```
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())
```

Στις Γραμμές 20 – 25, φαίνεται η έτοιμη βιβλιοθήκη που το λογισμικό εντοπίζει αντικείμενα. Με λίγα λόγια, είναι η ποικιλία των στόχων που θα εντοπίζει η κατασκευή μας. Το TheRaven όμως, θέλουμε να στοχοποιεί μόνο στρατιωτικά αντικείμενα, και γι’ αυτό με την εντολή IGNORE, αφήνουμε να εντοπίζονται μόνο αεροπλάνα και πλοία, δίνοντας έτσι τη δυνατότητα να χρησιμοποιηθεί σε Αντιαεροπορικό Πύραυλο ή σε Αντιαποβατικό Πυραυλικό σύστημα. Αξίζει να σημειωθεί, ότι η εντολή IGNORE ήταν το σημείο που προκάλεσε την προτίμηση του συγκεκριμένο πακέτου κώδικα, από αυτό του ‘Deep Learning Object Detection with OpenCV’.

```
# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
            "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
            "sofa", "train", "tvmonitor"]
IGNORE = set(["background", "bicycle", "bird", "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
             "dog", "horse", "motorbike", "person", "pottedplant", "sheep", "sofa", "train", "tvmonitor"])
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))
```

Στη συνέχεια, στις Γραμμές 33 – 38, ενεργοποιούμε τα Pins που θα δίνουν εντολή, ρεύμα και γείωση στο servomotor μας, αφού στον κορμό του κώδικα δεν περιήχετο σερβομηχανισμός. Σχετικά εύκολο εφόσον συνδέουμε στα φυσικά Pins του RaspberryPi το σερβομηχανισμό, και δεν χρησιμοποιούμε εξωτερική πηγή ρεύματος.

```
#define Servos GPIOs
GPIO.setmode(GPIO.BOARD)
servoPin=11
GPIO.setup(servoPin, GPIO.OUT)
pwm=GPIO.PWM(servoPin, 50)
pwm.start(7)
```

Το πιο δύσκολο κομμάτι όλου του κώδικα, ήταν στις Γραμμές 40 – 47. Εκεί, καθορίζεται η κίνηση του σερβομηχανισμού, αριστερά – δεξιά, εφόσον ο στόχος κινείται στην αντίστοιχη κατεύθυνση της οθόνης μας. Το συγκεκριμένο κομμάτι κώδικα, ενώ φαίνεται απλό, είναι κάτι που δεν υφίσταται πουθενά στο OpenSource Web, και εδώ έγκνεται η καινοτομία του project μας, καθόσον ότι υπάρχει από κίνηση Pan & Tilt, έχει να κάνει με τον εντοπισμό του χρώματος ή της θερμοκρασίας και όχι με το περίγραμμα (contour), το οποίο χρησιμοποιούμε εμείς στον κώδικά μας, και από εκεί και πέρα, στην εξαγωγή συντεταγμένων και την κίνηση αντιστοίχως των servomotors. Στην περίπτωση του TheRaven, εφόσον στις Γραμμές 99 – 102, υπολογίζονται οι συντεταγμένες, εκμεταλλευτήκαμε αυτές τις συντεταγμένες, οι οποίες όμως είναι για τον υπολογισμό των άκρων – σχηματισμό του 'τετραγώνου' που φαίνεται όταν εντοπιστεί ο στόχος μας. Επομένως, δεν είναι έτοιμες για 'εκμετάλλευση' καθόσον δεν έχουν την μορφή που πιστεύει κάποιος. ι συγκεκριμένες συντεταγμένες παίρνουν τιμές από 0 – 1 και πολλαπλασιάζονται με τα pixels για να εξαχθεί η κατασκευή του τετραγώνου. Με την δημιουργία των Γραμμών 40 – 47 καταφέραμε να κινείται ο σερβομηχανισμός μας σε γωνία 160° και 30° αναλόγως σε ποιο άκρο της οθόνης εντοπίζεται ο στόχος μας.

```
# position servo to "follow" the target
def mapServoPosition (startX):
    global panAngle
    if startX < 0.4:
        pwm.ChangeDutyCycle(7)
        time.sleep = (.05)
    else:
        pwm.ChangeDutyCycle(3)
        time.sleep = (.05)
```

Στις Γραμμές 50 – 52 εκκινούμε τη λειτουργία του σερβομηχανισμού μας από τις 90° , ούτως ώστε οι πτέρυγες να είναι σε παράλληλη με την κίνηση του πυραύλου, κατεύθυνση. Έπειτα, στις Γραμμές 54 – 56, εισάγουμε το τελευταίο μοντέλο μας, Caffé model, που

ουσιαστικά εισάγονται ως παράμετροι τα στοιχεία εντολών. Εκκινούμε την μετάδοση video (Γραμμές 58 – 63), δίνοντας χρόνο προθέρμανσης στην κάμερα, καθόσον η άμεση ενεργοποίησή της θα προκαλέσει φθορά των ηλεκτρολογικών της κυκλωμάτων, ακόμη κι του αισθητήρα της.

```
# Initialize angle servo at 90 position
global panAngle
panAngle = 90

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

# initialize the video stream, allow the camera sensor to warmup,
# and initialize the FPS counter
print("[INFO] starting video stream...")
vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)
fps = FPS().start()
```

Παρακάτω, στις Γραμμές 65 – 80, παίρνουμε ένα κυτίο εντοπισθέντος αντικειμένου, γίνεται προσαρμογή στο σχήμα του, και καταλήγουμε με προϊόντα το κυτίο, το είδος του στόχου και το ποσοστό επιτυχίας, χωρίς κάτι απ' αυτά να αποτυπώνεται στην οθόνη μας.

```
# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum width of 400 pixels
    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    # grab the frame dimensions and convert it to a blob
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
                                  0.007843, (300, 300), 127.5)

    # pass the blob through the network and obtain the detections and
    # predictions
    net.setInput(blob)
    detections = net.forward()
```

Στις Γραμμές 82 – 115, επεξεργαζόμαστε – εμεταλλευόμαστε τα δεδομένα που έχουμε, και εξετάζουμε τον κάθε στόχο που έχει ήδη ανιχνευθεί. Δηλαδή, ελέγχουμε την αξιοπιστία του εντοπισμού, και επιβεβαιώνουμε ότι βρίσκεται πάνω από ένα ποσοστό, και βλέπουμε εάν η κλάση του εντοπισμού μας έχει συμπεριληφθεί παραπάνω στην εντολή IGNORE, εάν ναι, τότε αγνοείται και συνεχίζεται κανονικά η διαδικασία για τον επόμενο στόχο. Στη συνέχεια, υπολογίζονται οι συντεταγμένες, οι οποίες έχουν πεδίο τιμών 0 – 1,



ούτως ώστε να κατασκευαστεί το κυτίο πέριξ του εντοπισθέντος στόχου, όπως και γίνεται αμέσως μετά. Τέλος, καλούμε την εντολή που ορίσαμε στις Γραμμές 41 – 48, και οι πτέρυγές μας κατευθύνονται πάνω στο στόχο.

```
# loop over the detections
for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with
    # the prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the 'confidence' is
    # greater than the minimum confidence
    if confidence > args["confidence"]:
        # extract the index of the class label from the
        # 'detections'
        idx = int(detections[0, 0, i, 1])

        # if the predicted class label is in the set of classes
        # we want to ignore then skip the detection
        if CLASSES[idx] in IGNORE:
            continue

        # compute the (x, y)-coordinates of the bounding box for
        # the object
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # draw the prediction on the frame
        label = "{}: {:.2f}%".format(CLASSES[idx],
                                    confidence * 100)
        cv2.rectangle(frame, (startX, startY), (endX, endY),
                      COLORS[idx], 2)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(frame, label, (startX, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

    # position Servo at target
    mapServoPosition (startX, startY, endX, endY)
```

Τελειώνοντας, στις Γραμμές 117 – 126, κλείνουμε το Loop, θέτοντας το Q σαν κομβίο διακοπής της διεργασίας, ανανεώνοντας τις πληροφορίες μας για τον μετρητή FPS. Παρακάτω, αποτυπώνουμε στην οθόνη μας τις πληροφορίες του μετρητή μας, απενεργοποιούμε τα GPIO και κάνουμε ένα cleanup, ενέργεια που αποσκοπεί στην ομαλότερη επανεκκίνηση όλων των περιφερειακών.

```
# show the output frame
cv2.imshow("Eikona Mou", frame)
key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop
if key == ord("q"):
    break

# update the FPS counter
fps.update()

# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# do a bit of cleanup
pwm.stop()
GPIO.cleanup()
cv2.destroyAllWindows()
vs.stop()
```

Γενικώς, ο κώδικας που χρησιμοποιήθηκε για την λειτουργία του TheRaven είναι σχετικά απλός, γεγονός απαραίτητο για τον διατιθέμενο επεξεργαστή του RaspberryPi, που ακόμη και έτσι, η απόκρισή του είναι με αρκετή καθυστέρηση.

./.



## ΚΕΦΑΛΑΙΟ 6

### Υλοποίηση Έργου

#### 1. Κατάσταση

Έχοντας διεκπεραιώσει όλες τις παραπάνω ενέργειες με επιτυχία το αποτέλεσμα που έχουμε από το TheRaven είναι η ομαλή λειτουργία του όσον αφορά την εικόνα και το βίντεο να απεικονίζεται πλήρως στην οθόνη με κάποια χρονοκαθυστέρηση της τάξης των 3 sec. Στη συνέχεια κατά την προβολή μίας μακέτας αεροπλάνου, βλέπουμε την στοχοποίησή του αφού διαγράφεται ένα κυττά γύρω του και φαίνεται το αποτέλεσμα της ανίχνευσης με το ποσοστό της εγκυρότητας πχ. “aeroplane 84%”. Στην κίνηση του στόχου προκαλείται μία διακοπή του παραπάνω εντοπισμού και μόλις σταθεροποιείται, περνάει στην προηγούμενη κατάσταση. Είναι γεγονός ότι και σ’ αυτή τη διεργασία το TheRaven παρουσιάζει μία αντίστοιχη χρονική καθυστέρηση. Τέλος, όταν ο στόχος βρίσκεται στο αριστερό τμήμα της οθόνης μας, οι πτέρυγες μετατοπίζονται δεξιά, γεγονός που θα μετέτρεπε την πορεία του βλήματος – πυραύλου πάνω στον οποίο θα είχε προσαρμοστεί το TheRaven.

Βλέπουμε ότι το εγχείρημα του TheRaven παρουσιάζει αποτελεσματικότητα ως προς τις απαιτήσεις του έργου, κάνοντας εμφανή την παρουσία υποδεέστερου επεξεργαστή από ότι απαιτείται, αφού όλα τα παραπάνω στοιχεία ελλιπούς απόδοσης αφορά ανεπάρκεια του επεξεργαστή και μόνο.

#### 2. Ποσοστό Ικανοποίησης Τεθέντος Έργου

##### 2.1 Hardware

Η ορθή επιλογή του υλικού το οποίο υλοποίησε το TheRaven ήταν μερικώς εύστοχο, καθόσον ο επεξεργαστής κρίνεται σχετικά ανεπαρκής για το επιθυμητό

αποτέλεσμα. Η χρησιμοποίηση ανώτερου υπολογιστή θα παρουσίαζε ένα TheRaven άλλου επιπέδου και προδιαγραφών...

## 2.2 Software

Το λογισμικό που χρησιμοποιήθηκε, από το Λειτουργικό Σύστημα μέχρι το αρχείο εκτέλεσης των διεργασιών του TheRaven στο σύνολό του, λειτούργησε ορθά επιτυγχάνοντας το σκοπό και τις προσδοκίες που υπήρχαν κατά την ανάθεση του εν λόγω έργου. Πιθανόν, να υπάρξει βελτιωμένη απόδοση, δοθέντος του συγκεκριμένου Hardware, εάν προγραμματιστεί σε κάποια γλώσσα προγραμματισμού διαφορετική της Python ή εφόσον χρησιμοποιηθεί κάποιο άλλο περιβάλλον Computer Vision.

## 2.3 Κατασκευή DIY

Η κατασκευή του συστήματος κατά το οποίο ο σερβομηχανισμός μετακινεί τον κατευθυντήριο μηχανισμό του TheRaven, είναι ευτελούς αξίας και τεχνολογίας, γεγονός που υποβιβάζει την αξία του ανατεθέντος έργου. Σε περίπτωση που χρησιμοποιούνταν υλικά βιομηχανικού επιπέδου και τεχνολογίας, το αποτέλεσμα θα έπειθε και τον πιο απαιτητικό χρήστη.

## 3. Δοκιμές

Μετά την ολοκλήρωση όλων των παραπάνω βημάτων που αφορούν τη δημιουργία του TheRaven, πρέπει να ακολουθήσουν πειράματα και δοκιμές, ούτως ώστε να επαληθευτούν οι προσδοκίες για την εν λόγω κατασκευή και να βελτιωθούν κάποιες λειτουργίες του. Από τις πολλαπλές δοκιμές που έγιναν για το TheRaven, υπό διάφορες προϋποθέσεις, εξήχθησαν κάποια συμπεράσματα, τα οποία παρατίθενται παρακάτω, κατά παράγοντα επιρροής:

### 3.1 Φωτισμός

Η ένταση του φωτός παίζει καθοριστικό ρόλο στο TheRaven, καθόσον η κάμερα δεν διαθέτει νυχτερινή όραση και έχει βασική ανάλυση εικόνας. Έτσι, με την παντελή έλλειψη φωτισμού δεν υπάρχει η δυνατότητα να ανιχνεύσει οτιδήποτε, πρόβλημα το οποίο επιλύεται, μόνο με την προσαρμογή και χρήση άλλης κάμερας, η οποία διαθέτει night vision. Με πλήρη φωτισμό, το TheRaven αντιλαμβάνεται πλήρως την παρουσία και την κίνηση ενός στόχου. Τον εντοπίζει, τον ακολουθεί και κινείται αναλόγως με την κίνησή του. Η ιδανική περίπτωση όμως, κατά την οποία το TheRaven κερδίζει, εκτός από το αποτέλεσμα και σε χρόνο, είναι αυτή που ενώ υπάρχει συσκοτίσιμη περιβάλλοντος, ο στόχος φωτίζεται από κάποια εξωτερική πηγή, καθιστώντας τον το μοναδικό δεδομένο που έχει να επεξεργαστεί το TheRaven και έτσι, ο χρόνος αντίδρασης σε όλα τα στάδια της λειτουργίας του είναι ελαττωμένος.

### 3.2 Στόχος

Σε όλα τα πειράματα χρησιμοποιήθηκε μία μακέτα αεροσκάφους, η οποία απεικόνιζε το στόχο και κινούνταν στο πεδίο παρατήρησης του TheRaven, για να προκαλέσει την λειτουργία του. Στην περίπτωση, κατά την οποία, ο στόχος ήταν στηριγμένος από ένα ψηλότερο σημείο και κρεμόταν με αόρατο για το TheRaven νήμα, τότε, ο εντοπισμός του γινόταν κανονικά και το κυτίο σχηματιζόταν ακριβώς στις διαστάσεις του TheRaven. Αντιθέτως, όταν κρατούσαμε με το χέρι το αεροσκάφος, συνέβη να μην μπορεί να ανιχνευθεί ο στόχος, καθόσον το χέρι θεωρούταν σώμα του στόχου. Επίσης, συνέβη, κατά την ίδια περίπτωση, να περικλείεται, μέσα στο κυτίο, και το χέρι που κρατούσε το αεροσκάφος...

### 3.3 Απόσταση

Ένας άλλος σημαντικός παράγοντας που επηρεάζει την απόδοση του TheRaven, είναι η απόσταση. Σε απόσταση μικρότερη του ενός (1) μέτρου, ο στόχος παίρνει μεγάλες διαστάσεις, με αποτέλεσμα το TheRaven να θεωρεί ότι είναι στη σωστή κατεύθυνση και δεν αλλάζει στόχο. Αντίθετα, σε απόσταση μεγαλύτερη των 20 μέτρων, δεν είναι εύκολα αντιληπτός ο στόχος, καθόσον το μήκος του αντιστοιχεί περίπου στο 10% του μήκους του εύρους που 'βλέπει' το TheRaven. Έτσι, η παραπάνω αναλογία πρέπει να είναι μεταξύ 15 – 30%, ούτως ώστε να λειτουργεί η κατασκευή μας ικανοποιητικά.

./.

## ΚΕΦΑΛΑΙΟ 7

### What else? What more?

#### 1. Εναλλακτική Τεχνολογία - Προοπτική

Αξίζει να αναφερθεί η υπάρχουσα τεχνολογία που διατίθεται σήμερα στα οπλικά συστήματα της παγκόσμιας βιομηχανίας αμυντικών συστημάτων, όπως αυτή του F-22 Raptor. Πιο συγκεκριμένα, ο πύραυλος που διαθέτει το συγκεκριμένο Α/Φ, Sidewinder είναι σχεδιασμένος να εκρήγνυται λίγα μέτρα πριν το στόχο του και γι' αυτό διαθέτει ένα σύστημα οπτικού εντοπισμού του στόχου, ούτως ώστε να γνωρίζει ο πύραυλος πότε βρίσκεται εντός εμβελείας. Ο εντοπιστής αυτός αποτελείται από 8 διόδους λέιζερ και 8 διόδους που λειτουργούν με αισθητήρα φωτός. Έτσι, όταν βρίσκεται εν πτήση ο πύραυλος, οι δίοδοι λέιζερ στέλνουν συνεχώς δέσμες και όταν κάποια επιστρέψει γρήγορα, σημαίνει ότι ο στόχος βρίσκεται εντός εμβελείας και δίνεται η εντολή για έκρηξη του πυραύλου. Με λίγα λόγια, βλέπουμε ένα σύστημα με διπλό τρόπο λειτουργίας, για εξασφάλιση του αποτελέσματος, με τρόπο ενεργείας σχετικά απλό και προηγούμενων τεχνολογιών (sonar).

Η χρήση του OpenCV και η κατασκευή του TheRaven θα μπορούσε να γίνει ούτως ώστε να εντοπίζει το στόχο ανιχνεύοντας το χρώμα και όχι το περίγραμμα. Τότε θα ήταν εφικτή η εύρεση μεγάλου όγκου πληροφορίας καθόσον το μήκος κύματος του φωτός που εκπέμπει ένα σώμα, το χρώμα του δηλαδή, είναι κάτι που εύκολα ανιχνεύεται και είναι μία τεχνική που έχει χρησιμοποιηθεί αρκετά την τελευταία δεκαετία στα αμυντικά συστήματα. Όμως, δεν χρησιμοποιήθηκε αυτός ο δρόμος στο TheRaven, διότι λήφθηκε υπ' όψιν ότι τα μέτρα παθητικής ασφάλειας που έχουν αναπτυχθεί σαν απάντηση στα παραπάνω, όπως χρώματα βαφής anti-IR, είναι πλέον πολύ αποτελεσματικά και η ανάγκη για εξέλιξη αυτών των συστημάτων εντοπισμού είναι ολοένα και μεγαλύτερη. Βέβαια, ο τρόπος λειτουργίας του TheRaven έχει περιορισμένες δυνατότητες, καθόσον η βιβλιοθήκη των περιγραμμάτων από

τα οποία ‘διαλέγουμε’ το είδος του στόχου, πρέπει να είναι μικρή σε αριθμό, κι αυτό διότι ο όγκος των δεδομένων αυξάνεται αισθητά και η λειτουργία του επεξεργαστή επιβραδύνεται.

Η αποτελεσματικότητα του TheRaven μπορεί να χαρακτηριστεί ως μέτρια και αυτό οφείλεται στο hardware που χρησιμοποιήθηκε.

Εάν όμως χρησιμοποιούνταν κάποιος πιο εξεζητημένος, αποτελεσματικός και γρήγορος επεξεργαστής το αποτέλεσμα θα ήταν σαφώς πιο ικανοποιητικό. Όπως για παράδειγμα ο επεξεργαστής iQ7 (Εικ.26)..



(Εικ.26)

Κατά τον ίδιο τρόπο, εάν η χρησιμοποιούμενη κάμερα ήταν υψηλότερων προδιαγραφών, μεγαλύτερης ευκρίνειας, υψηλής ανάλυσης και με δυνατότητα οπτικής και ψηφιακής εστίασης, θα είχε τη δυνατότητα το TheRaven να ανιχνεύσει, εντοπίσει και εστιάσει σε στόχο που βρίσκεται σε μεγάλες αποστάσεις και σε συνθήκες μειωμένης ορατότητας. Μία τέτοια κάμερα είναι για παράδειγμα της εταιρείας Lumenera Corporation (Εικ.27).

MACHINE VISION CAMERA / INFRARED / CCD / USB 3.0  
LT365R, LT665R, LT965R, LT1265R



[Εικ.27]

Περιθώρια βελτίωσης και κομμάτια του TheRaven που επιδέχονται τροποποίηση υπάρχουν, όπως για παράδειγμα η κατεύθυνση του βλήματος. Θα ήταν πολύ πιο ρεαλιστικό το παρόν σύστημα προώθησης, που διαθέτει ένα σταθερό κινητήριο ή καθόλου και η κατεύθυνση του πυραύλου – βλήματος γίνεται με πτέρυγες, να υπήρχε ένας πυραυλοκινητήρας, ο οποίος θα κατευθυνόταν ο ίδιος σαν την βάση ενός joystick δηλαδή. Βέβαια, η ανάγκη του επεξεργαστή για περισσότερες διεργασίες ταυτόχρονα θα ήταν πολλαπλάσια.

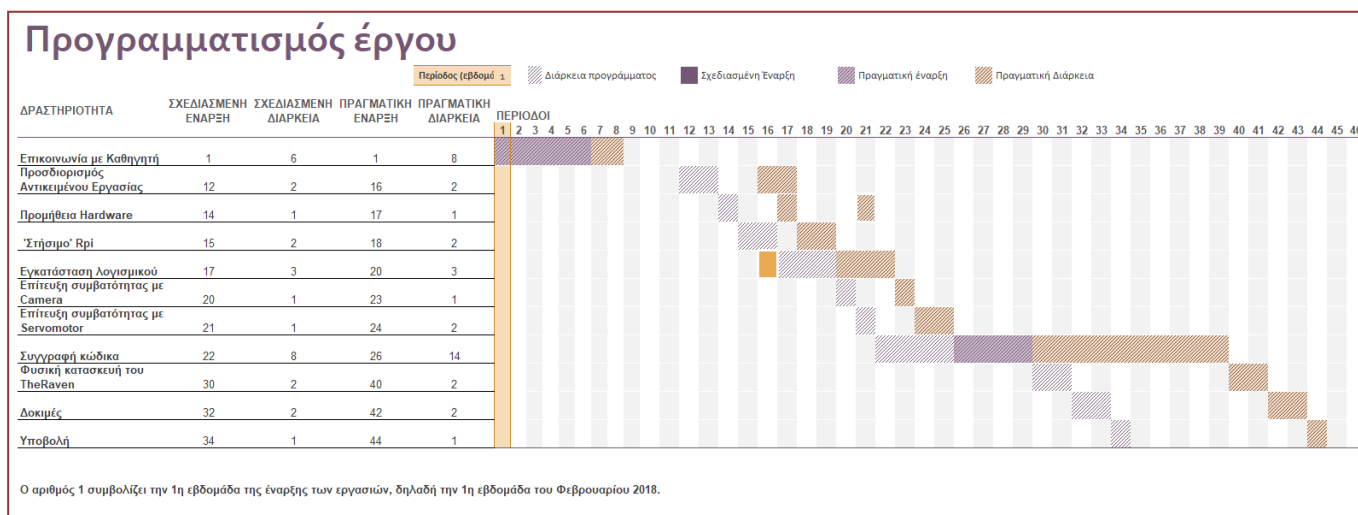
Έτσι, θα είχαμε τεράστια ακρίβεια αλλά και δυνατότητα εύκολης και γρήγορης αλλαγής κατεύθυνσης.

Επίσης, για την πληρέστερη λειτουργία του project θα ήταν καλό να υπήρχε σε λειτουργία και ένα γυροσκόπιο μαζί με ένα επιταχυνσιόμετρο, δεδομένα των οποίων θα συμπεριλαμβάνονταν στον κώδικα, και θα είχε πληρέστερη εικόνα της κίνησης, της θέσης και της κατεύθυνσης το TheRaven. Με τον χρησιμοποιούμενο επεξεργαστή, όμως, το αποτέλεσμα του επεξεργαστή θα ήταν λιγότερο από ανεπαρκές...

Τέλος, για την ρεαλιστικότερη κατασκευή του ίδιου του φορέα που φαίνεται στην παρ.4, θα ήταν προτιμότερο να είχαν χρησιμοποιηθεί υλικά, εργαλεία και μεθοδολογία βιομηχανικού επιπέδου..

## 2. Χρονικές Λεπτομέρειες Έργου

Ξεκινώντας την προσπάθεια δημιουργίας του TheRaven, έγινε ένα χρονοδιάγραμμα, ένα πλάνο δηλαδή για του τι επιμέρους στάδια θα ακολουθηθούν και πόσο εκτιμάται ότι θα διαρκέσουν αυτά. Η εκτίμηση έγινε με βάση τις προσωπικές γνώσεις και το είδος του εκάστοτε βήματος διεκπεραίωσης της εργασίας. Παρακάτω, φαίνεται ο εν λόγω σχεδιασμός, αλλά και η πραγματική χρονική διάρκεια του κάθε βήματος της εργασίας.



Όπως φαίνεται παραπάνω, η εκτίμηση της χρονικής διάρκειας, σε πολλές περιπτώσεις, απέχει από την πραγματική χρονική διάρκεια που απαιτήθηκε για περάτωση του κάθε βήματος – σταδίου της επιχείρησης. Ένας λόγος, για τον οποίο συνέβη αυτό, είναι το γεγονός ότι η εκτίμηση δεν έγινε με πλήρη γνώση των απαιτήσεων ορισμένων σταδίων, όπως η συγγραφή του κώδικα.

Συμπερασματικά, θα έπρεπε να ακολουθηθεί πιο οργανωμένη μεθοδολογία κατά την πρόβλεψη και εκτίμηση της χρονικής διάρκειας διεκπεραίωσης του έργου.

./.



## ΕΠΙΛΟΓΟΣ

Με λίγα λόγια, όλα τα βήματα και οι εργασίες που πραγματοποιήθηκαν, αναφέρθηκαν παραπάνω, με κάθε λεπτομέρεια, με απώτερο σκοπό πάντα τη δημιουργία και κατασκευή του TheRaven, μηχανισμός ο οποίος δύναται να λειτουργήσει σαν απάχτιο ενός αμυντικού συστήματος και να βελτιώσει την αποτελεσματικότητά του.

Βέβαια, κάποιος δεν μπορεί να διακρίνει το πέρας της λειτουργίας, γεγονός που δεν φαίνεται στον προγραμματισμό του TheRaven, καθόσον είναι πακτωμένο πάνω στο βλήμα – πύραυλο και εκτελεί τις διεργασίες του μέχρι την πρόσκρουσή του πάνω στο στόχο, οπότε και καταστρέφεται....

Η παραπάνω προσπάθεια αποτελεί ένα παράδειγμα ότι η τάση του πληθυσμού να έχει επιρροή στην τεχνολογία που αναπτύσσεται γύρω του είναι ολοένα αυξανόμενη, καθόσον με στοιχειώδεις γνώσεις πληροφορικής, προγραμματισμού, μικροηλεκτρονικών και τεχνολογίας όπλων, μπορεί κάποιος να καταφέρει πολλά περισσότερα σε σχέση με το παρελθόν. Αυτό συμβαίνει διότι η τεχνολογία είναι κομμάτι της καθημερινότητας, η πρόσβαση στη γνώση και την πληροφορία είναι ελεύθερη και δωρεάν και τέλος, η τεχνολογία έχει απλουστευθεί με στόχο την απόκτηση μεγαλύτερου κοινού. Όπως διαφαίνεται, στο μέλλον οποιοσδήποτε θα έχει τη δυνατότητα να ελέγχει, να προγραμματίζει, να τροποποιεί και να καθορίζει επακριβώς και εις βάθος όλες τις διεργασίες που θα συμβαίνουν συνεχώς, γύρω του, από τις εργασίες και τη λειτουργία του νοικοκυριού, των διαφόρων οχημάτων του, αλλά και την εργασία του, που θα είναι ψηφιοποιημένα σε ένα μεγάλο ποσοστό.

./.

**ΒΙΒΛΙΟΓΡΑΦΙΑ**

1. [The](#) MagPi, “Beginner’s Guide to NOOBS”, *Beginner’s Guide*, January 2018.
2. [The](#) MagPi, “The Pi Camera”, *Beginner’s Guide*, January 2018.
3. [The](#) MagPi, “Code in Python with Thony”, *Beginner’s Guide*, January 2018.
4. [The](#) MagPi, “Power your RaspberryPi”, *Beginner’s Guide*, January 2018.
5. [The](#) MagPi, “The Pi Camera”, *Beginner’s Guide*, January 2018.
6. [The](#) MagPi, “RaspberryPi 3B+”, *RaspberryPi 3B+*, April 2018.
7. [The](#) MagPi, “Model Railway”, *Projects Book*, January 2018.
8. [The](#) MagPi, “Discoverer”, *Projects Book*, January 2018.
9. [The](#) MagPi, “Pi Motor Shim”, *Projects Book*, January 2018.
10. Doxygen, “Introduction to OpenCV”, <https://docs.opencv.org/master>, January 2018.
11. Rosebrock, Adrian, “A gentle guide to deep learning object detection”, <https://www.pyimagesearch.com/2018/05/14/>, May 2018.
12. Rosebrock, Adrian, “Deep learning object detection with OpenCV”, <https://www.pyimagesearch.com/2017/10/16/>, October 2017.
13. Σχολή Ναυτικών Δοκίμων, «Πυρομαχικά», *Βλήματτα, Πύραυλοι*, Ιούνιος 2015.
14. Jeffrey, Thomas, “Performance – driven Requirements”, *Fundamentals of Systems Engineering for Defence Systems Applications*, June 2017.
15. Jeffrey, Thomas, “System Performance Evaluation”, *Fundamentals of Systems Engineering for Defence Systems Applications*, June 2017.
16. Jeffrey, Thomas, “Architecting Projects”, *Fundamentals of Systems Engineering for Defence Systems Applications*, June 2017.
17. Jeffrey, Thomas, “Systems Engineering Tools”, *Fundamentals of Systems Engineering for Defence Systems Applications*, June 2017.
18. Jeffrey, Thomas, “Use of Leading Indicators”, *Fundamentals of Systems Engineering for Defence Systems Applications*, June 2017.
19. Systems Management College, “Systems Engineering Process Overview”, *Systems Engineering Fundamentals*, January 2001.
20. Systems Management College, “Design Synthesis”, *Systems Engineering Fundamentals*, January 2001.

21. Systems Management College, “Work Breakdown Structure”, *Systems Engineering Fundamentals*, January 2001.
22. Systems Management College, “Risk Management”, *Systems Engineering Fundamentals*, January 2001.
23. Systems Management College, “Systems Engineering Planning”, *Systems Engineering Fundamentals*, January 2001.
24. Systems Management College, “Organizing and Integrating System Development”, *Systems Engineering Fundamentals*, January 2001.
25. Systems Management College, “Configuration Management”, *Systems Engineering Fundamentals*, January 2001.
26. Systems Management College, “Verification”, *Systems Engineering Fundamentals*, January 2001.
27. Systems Management College, “Systems Engineering Process Output”, *Systems Engineering Fundamentals*, January 2001.
28. Monk Simon, “Using a Servomotor with RaspberryPi”, *Adafruit* ©, October 2016.
29. MJRoBot, “Pan- Tilt Multi Servo Control”, *Hackster* ©, February 2018.
30. MJRovai, “Automatic Vision Object Tracking”, *Instructables* ©, February 2018.