



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Διπλωματική Εργασία

Σχεδίαση και υλοποίηση συστήματος
ρομποτικής κατασκευής με υποστήριξη
μηχανικής όρασης

Design and implementation of robotic system
supported by computer vision

Χασχατζή Ακριβή

Χατζησάββας Αντώνιος

Επιβλέπων Καθηγητής: **Δρ. Μηνάς Δασυγένης**

Εργαστήριο Ρομποτικής, Ενσωματωμένων & Ολοκληρωμένων Συστημάτων

Κοζάνη, Οκτώβριος 2020

Περιεχόμενα

Περιεχόμενα	3
Κατάλογος εικόνων	7
Κατάλογος πινάκων.....	11
Περίληψη.....	13
Abstract.....	15
Δήλωση Πνευματικών Δικαιωμάτων	17
Ευχαριστίες.....	19
Κεφάλαιο 1ο – Εισαγωγή.....	21
1.1 Σημερινή Κατάσταση	21
1.2 Σκοπός του έργου.....	22
1.3 Παρόμοιες Εργασίες	22
1.3.1 Moxie	23
1.3.2 QTrobot	24
1.3.3 Σύγκριση	25
1.4 Σύνοψη.....	26
Κεφάλαιο 2ο – Θεωρητικό Υπόβαθρο	27
2.1 Λογισμικά - Εργαλεία.....	27
2.1.1 Python.....	27
2.1.2 OpenCV Python	28
2.1.3 PyQt v5.....	29

2.1.4 Qt Creator.....	31
2.1.5 GCompris	32
2.1.6 MobaXterm	33
2.1.7 Nano	34
2.1.8 Notepad++.....	35
2.1.9 Visual Studio Code.....	36
2.1.10 Fusion 360.....	37
2.1.11 Prusa i3 mk3.....	38
2.2 Υλικός εξοπλισμός.....	40
2.2.1 Raspberry Pi	40
2.2.2 Υποβιβαστής τάσης HW-319-v6.0	42
2.2.3 Βηματικοί κινητήρες και οδηγός βηματικών κινητήρων	42
2.2.4 Σέρβο κινητήρες και οδηγός σέρβο κινητήρων	44
2.2.5 Χωρητικός αισθητήρας αφής (Capacitive Touch Sensor) HW-139	45
2.2.6 Αισθητήρας απόστασης υπερήχων HY-SRF05	46
2.2.7 Αισθητήρας Υπέρυθρων HW-006 V1.3.....	47
2.2.8 Δίοδοι εκπομπής φωτός (LED) σε σχήμα στεφάνης 12 στοιχείων WS2812B	49
2.3 Σύνοψη.....	50
Κεφάλαιο 3 – Δομικά μέρη συστήματος.....	51
3.1 Απαιτήσεις	51
3.2 Περιπτώσεις χρήσης	52
3.3 Υλικό μέρος	53

3.4 Συνδεσμολογία.....	54
3.5 Ενεργειακές απαιτήσεις	57
3.6 Σχεδιασμός και εμφάνιση ρομπότ	58
3.7 Σύνοψη.....	59
Κεφάλαιο 4 – Ανάλυση Λογισμικού.....	61
4.1 Λογισμικό μικροεπεξεργαστή.....	61
4.2 Συναρτήσεις.....	62
4.2.1 Κουμπί πολλαπλών χρήσεων	62
4.2.2 Αισθητήρες απόστασης.....	62
4.2.3 Επαγωγικοί αισθητήρες αφής.....	64
4.2.4 Δίοδοι εκπομπής φωτός.....	64
4.2.5 Σέρβο κινητήρες.....	65
4.2.6 Βηματικοί κινητήρες.....	66
4.3 Διασύνδεση με πληροφοριακό σύστημα	67
4.4 Γραφικά QtDesigner	69
4.5 Βασική συνάρτηση PyQt v5	78
4.6 Προ εγκατεστημένα σενάρια	86
4.6.1 Σενάριο «Ειδική Ενσυναίσθηση»	87
4.6.2 Σενάριο «Ειδική Κοινωνική Αλληλεπίδραση»	91
4.6.3 Σενάριο «Εκφοβισμός»	95
4.6.4 Σενάριο «Κοινωνική Συμπεριφορά».....	97
4.6.5 Σενάριο «Επίδειξη».....	100

4.7 Σύνοψη.....	101
Κεφάλαιο 5	103
5.1 Σύνοψη.....	103
5.2 Μετρικά Συστήματος.....	104
5.3 Ανάλυση SWOT	104
5.3.1 Δυνατά Σημεία	104
5.3.2 Αδυναμίες.....	105
5.3.3 Ευκαιρίες.....	105
5.3.4 Απειλές.....	105
5.4 Ασφάλεια	106
5.5 Μελλοντικές Επεκτάσεις	106
5.6 Προβλήματα που αντιμετωπίστηκαν	107
5.7 Στάδια υλοποίησης	108
Βιβλιογραφία	109
Παράρτημα Α – Εγκατάσταση OpenCV	111
Βήμα 1: Ανάπτυξη του συστήματος αρχείων και ανάκτηση χώρου.....	111
Βήμα 2: Εγκατάσταση εξαρτήσεων.....	112
Βήμα 3: Δημιουργία του εικονικού περιβάλλοντος Python και εγκατάσταση του NumPy....	113
Βήμα 4: Εγκατάσταση OpenCV 4 με την εντολή pip	114
Βήμα 5: Δοκιμή της εγκατάστασης OpenCV 4 Raspberry Pi BusterOS.....	114
Παράρτημα Β – Εγκατάσταση βιβλιοθηκών ήχου, του PyQt και του gcompris	115
Παράρτημα Γ – Εγκατάσταση βιβλιοθηκών για LED, κινητήρες και WiFi	117

Κατάλογος εικόνων

Εικόνα 1: Moxie εκπαιδευτικό ρομπότ (Embodied - https://embodied.com/).....	24
Εικόνα 2: QTrobot εκπαιδευτικό ρομπότ (luxai - https://luxai.com/).....	25
Εικόνα 3: Παράδειγμα κώδικα σε Python με το λογότυπο της γλώσσας.....	28
Εικόνα 4: OpenCV αναγνώριση αντικείμενων	29
Εικόνα 5: Λογότυπο PyQt	30
Εικόνα 6: Περιβάλλον Qt Creator	32
Εικόνα 7: Στιγμιότυπο αρχικής οθόνης GCompris	33
Εικόνα 8: Παράδειγμα σύνδεσης SSH σε MobaXterm με το λογότυπο του προγράμματος.....	34
Εικόνα 9: Παράδειγμα του κειμενογράφου Nano	35
Εικόνα 10: Παράδειγμα του κειμενογράφου Notepad++.....	36
Εικόνα 11: Περιβάλλον Visual Studio Code.....	37
Εικόνα 12: Περιβάλλον Fusion 360	38
Εικόνα 13: Εκτυπωτής Prusa i3 mk3.....	39
Εικόνα 14: Raspberry Pi 4.....	41
Εικόνα 15: Οθόνη αφής 7".....	41
Εικόνα 16: Raspberry Pi Camera	42
Εικόνα 17: Ρυθμιστής Τάσης HW-319-v6	42
Εικόνα 18: Εσωτερική κατανομή βηματικού κινητήρα	43
Εικόνα 19: Βηματικός κινητήρας 42BYGHW811X1	43

Εικόνα 20: Οδηγός βηματικών κινητήρων TB6560.....	44
Εικόνα 21: Servo MG995.....	45
Εικόνα 22: Servo controller PCA9685.....	45
Εικόνα 23: Αναπαράσταση παρεμβολής αντικείμενου σε μαγνητικό πεδίο.....	46
Εικόνα 24: Touch sensor HW-139.....	46
Εικόνα 25: Απεικόνιση πρόσκρουσης υπερηχητικού κύματος σε αντικείμενο.....	47
Εικόνα 26: Ultrasonic HY-SRF05.....	47
Εικόνα 27: Απεικόνιση πρόσκρουσης υπέρυθρου φωτός σε επιφάνεια και στο κενό.....	48
Εικόνα 28: Infrared HW-006.....	49
Εικόνα 29: LED ring WS2812B.....	49
Εικόνα 30: Διάγραμμα περιπτώσεων χρήσης.....	53
Εικόνα 31: Κυκλωματική απεικόνιση του συστήματος.....	54
Εικόνα 32: Είσοδοι/ έξοδοι μικροεπεξεργαστή.....	57
Εικόνα 33: Τρισδιάστατο σχέδιο εμφάνισης ρομπότ.....	59
Εικόνα 34: Συναρτήσεις ελέγχου υπερηχητικών αισθητήρων.....	63
Εικόνα 35: Συναρτήσεις ελέγχου υπέρυθρων αισθητήρων.....	63
Εικόνα 36: Συνάρτηση ελέγχου επαγωγικών αισθητήρων αφής.....	64
Εικόνα 37: Συνάρτηση διόδων εκπομπής φωτός.....	64
Εικόνα 38: Συνάρτηση ελέγχου σέρβο κινητήρων.....	65
Εικόνα 39: Αντιστοιχία σέρβο κινητήρων.....	65
Εικόνα 40: Συναρτήσεις κίνησης του ρομπότ μπροστά/πίσω.....	66
Εικόνα 41: Συναρτήσεις κίνησης του ρομπότ δεξιά/αριστερά.....	67

Εικόνα 42: Συνάρτηση RobotCheckConnection	68
Εικόνα 43: Λειτουργία πρωτοκόλλου HTTP	68
Εικόνα 44: Χαρακτηριστικά αντικειμένων QAction.....	70
Εικόνα 45: Στοιχεία γραμμής εργαλείων	70
Εικόνα 46: Δομή διεπαφών και γραμμής εργαλείων.....	70
Εικόνα 47: Στιγμιότυπο διεπαφής αρχικής οθόνης	71
Εικόνα 48: Στιγμιότυπο διεπαφής επιλογής κατηγορίας σεναρίων.....	72
Εικόνα 49: Στιγμιότυπο διεπαφής παρουσίασης προ εγκατεστημένων σεναρίων	73
Εικόνα 50: Στιγμιότυπο διεπαφής ρυθμίσεων δικτύου	76
Εικόνα 51: Στιγμιότυπα διεπαφών προ εγκατεστημένων σεναρίων	77
Εικόνα 52: Δήλωση γραφικών στην βασική συνάρτηση	79
Εικόνα 53: Συναρτήσεις ενημέρωσης κατάστασης σύνδεσης	80
Εικόνα 54: Συναρτήσεις γραμμής εργαλείων.....	81
Εικόνα 55: Συναρτήσεις εμφάνισης και εκτέλεσης των αντικείμενων αρχικής οθόνης	81
Εικόνα 56: Συναρτήσεις εύρεσης διευθύνσεων δικτύου	82
Εικόνα 57: Κλάση ρυθμίσεων δικτύου του συστήματος	83
Εικόνα 58: Οδηγίες σύνδεσης στο διαδίκτυο	83
Εικόνα 59: Συναρτήσεις εκτέλεσης προ εγκατεστημένων και σεναρίων χρήστη.....	84
Εικόνα 60: Συναρτήσεις επιστροφής και τερματισμού	85
Εικόνα 61: Συναρτήσεις εκτελέσεις προ εγκατεστημένων σεναρίων	86
Εικόνα 62: Η προσομοίωση των πέντε συναισθηματικών καταστάσεων	87
Εικόνα 63: Κουμπιά αλλαγής διάθεσης του ρομπότ	87

Εικόνα 64: Παράδειγμα εντοπισμού προσώπου.....	89
Εικόνα 65: Συνάρτηση εντοπισμού προσώπου σεναρίου «Ειδική Ενσυναίσθηση»	89
Εικόνα 66: Κλάση σεναρίου «Ειδική Ενσυναίσθηση»	90
Εικόνα 67: Συνάρτηση κινηματικού μοντέλου σεναρίου «Ειδική Ενσυναίσθηση»	90
Εικόνα 68: Συναρτήσεις αλλαγής διάθεσης σεναρίου «Ειδική Ενσυναίσθηση»	91
Εικόνα 69: Ενδεικτικές κάρτες ζώων σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»	92
Εικόνα 70: Αυτοκόλλητα σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση».....	93
Εικόνα 71: Κώδικας αναγνώρισης ζώων σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»	93
Εικόνα 72: Κλάση σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση».....	94
Εικόνα 73: Συνάρτηση αλλαγής κατάστασης σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»...	95
Εικόνα 74: Κλάση σεναρίου «Εκφοβισμός».....	96
Εικόνα 75: Συναρτήσεις αλλαγής κατάστασης και κίνηση σεναρίου «Εκφοβισμός»	97
Εικόνα 76: Κλάση σεναρίου «Κοινωνική Συμπεριφορά».....	98
Εικόνα 77: Συναρτήσεις σεναρίου «Κοινωνική Συμπεριφορά»	99
Εικόνα 78: Συναρτήσεις αλλαγής διάθεσης και κίνηση σεναρίου «Κοινωνική Συμπεριφορά»...	99
Εικόνα 79: Κινηματικό μοντέλο σεναρίου «Επίδειξη».....	100
Εικόνα 80: Κλάση σεναρίου «Επίδειξη».....	101
Εικόνα 81: Συνάρτηση λήψης φωτογραφιών σεναρίου «Επίδειξη»	101
Εικόνα 82: Οθόνη διαμόρφωσης του Rasbian Buster (raspi-config).....	111

Κατάλογος πινάκων

Πίνακας 1: Χαρακτηριστικά μικροεπεξεργαστή και των περιφερειακών του	42
Πίνακας 2: Χαρακτηριστικά HW-319-v6	42
Πίνακας 3: Χαρακτηριστικά βηματικού κινητήρα και οδηγού κινητήρων	44
Πίνακας 4: Χαρακτηριστικά σερβομηχανισμού και του οδηγού PCA9685	45
Πίνακας 5: Χαρακτηριστικά χωρητικού αισθητήρα αφής HW-139	46
Πίνακας 6: Χαρακτηριστικά υπερηχητικού αισθητήρα HY-SRF05	47
Πίνακας 7: Χαρακτηριστικά υπέρυθρου αισθητήρα HW-006	49
Πίνακας 8: Χαρακτηριστικά δίοδων εκπομπής φωτός σε στεφάνη WS2812B	49
Πίνακας 9: Πίνακας εισόδων/ εξόδων μικροεπεξεργαστή	57
Πίνακας 10: Διεπαφή αρχικής οθόνης	71
Πίνακας 11: Διεπαφή επιλογής κατηγορίας σεναρίων	72
Πίνακας 12: Διεπαφή παρουσίασης προ εγκατεστημένων σεναρίων	73
Πίνακας 13: Διεπαφή προ εγκατεστημένου σεναρίου «Ειδική Ενσυναίσθηση»	74
Πίνακας 14: Διεπαφή προ εγκατεστημένου σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»	74
Πίνακας 15: Διεπαφή προ εγκατεστημένου σεναρίου «Εκφοβισμός»	74
Πίνακας 16: Διεπαφή προ εγκατεστημένου σεναρίου «Κοινωνική Συμπεριφορά»	75
Πίνακας 17: Διεπαφή προ εγκατεστημένου σεναρίου «Επίδειξης»	75
Πίνακας 18: Διεπαφή ρυθμίσεων δικτύου	76
Πίνακας 19: Εσωτερική διεπαφή απεικόνισης εκφράσεων προσώπου	78
Πίνακας 20: Εσωτερική διεπαφή απεικόνισης ζώων	78

Περίληψη

Στα πλαίσια της σύγχρονης κοινωνίας, είναι απαραίτητο να κατανοήσουμε την σημαντικότητα της ενσυναίσθησης στην ζωή μας. Διαπιστώνεται από έρευνες ότι υπάρχει η ανάγκη ανάπτυξης πλατφορμών, οι οποίες είναι σε θέση να παράγουν συναισθήματα, προκειμένου παιδιά και ενήλικες να μάθουν να κατανοούν τα συναισθήματα των ανθρώπων που συναναστρέφονται.

Η συγκεκριμένη εργασία αποσκοπεί στην δημιουργία ενός ρομποτικού συστήματος, το οποίο θα μπορεί να χρησιμοποιηθεί τόσο από παιδιά ηλικίας τεσσάρων με εννέα ετών για να μάθουν παίζοντας, όσο και από χρήστες μεγαλύτερης ηλικίας. Τα παιδιά αξιοποιώντας τις λειτουργίες του ρομπότ είναι σε θέση να αναπτύξουν την ενσυναίσθηση τους. Στους ενήλικους χρήστες δίνεται η δυνατότητα ανάπτυξης και δημιουργίας των δικών τους προγραμμάτων, μιας και το ρομπότ είναι δομημένο με ανοιχτά πρότυπα (open source).

Έχουν αναπτυχθεί, με την βοήθεια ειδικών, σενάρια μέσα από το οποία το ρομπότ είναι σε θέση να εκφράζει συναισθήματα. Οι χρήστες έχουν την δυνατότητα να εκτελέσουν βασικά σενάρια, να ανεβάσουν με την βοήθεια πληροφοριακού συστήματος τα δικά τους προγράμματα, να παίξουν εκπαιδευτικά παιχνίδια και να χειριστούν το ρομπότ απομακρυσμένα μέσω ειδικής εφαρμογής τηλεχειρισμού.

Το σώμα του ρομπότ αποτελείται από πλαστικά κομμάτια εκτυπωμένα σε τρισδιάστατο εκτυπωτή. Στην κατασκευή έχουν τοποθετηθεί αισθητήρια όργανα, με τα οποία εξασφαλίζεται η ασφάλεια των χρηστών και οι απαιτήσεις των σεναρίων.

Το ρομποτικό σύστημα αναπτύχθηκε με την χρήση των πιο σύγχρονων μεθόδων. Τα γραφικά της οθόνης υλοποιήθηκαν με PyQt πέμπτης έκδοσης και ο προγραμματισμός των λειτουργιών του συστήματος με την χρήση της γλώσσας προγραμματισμού Python τρίτης έκδοσης. Η ανάπτυξη κώδικα πραγματοποιήθηκε, μέσω των εργαλείων MobaXterm, Nano, Notepad++, Visual Studio Code, Qt Creator, Qt Designer, ενώ ο σχεδιασμός και η μελέτη, των τρισδιάστατων τμημάτων της εργασίας, έγινε με το πρόγραμμα σχεδίασης Fusion 360.

Λέξεις κλειδιά: Ρομποτικό σύστημα, Ανάπτυξη ενσυναίσθησης, Μηχανική όραση, Raspberry Pi, PyQt5, Λογισμικό ανοιχτού κώδικα, Τρισδιάστατη εκτύπωση.

Abstract

In modern society, we must understand the importance of empathy in our lives. Research shows that there is a need to develop platforms that can generate emotions so that children and adults learn to understand the feelings of the people they associate with.

The main focus of this paper is to create a robotic system that can be used by children from the age of four to nine years old, to learn through playing, as well as by older users. Using the functions of the robot, children can develop their empathy. Adult users are allowed to develop and create their programs since the robot is structured with open standards (open source).

With the help of experts, scenarios have been developed in which the robot can express emotions. Users can run basic scripts, upgrade their programs with the help of an information system, play educational games, operate the robot remotely through a special remote-control application.

The body of the robot consists of plastic pieces printed on a 3D printer. The sensory equipment has been installed, ensuring the safety of the users and the requirements of the scenarios.

The robotic system was developed using the latest methods. The on-screen graphics were implemented with PyQt Fifth Edition and the programming of system functions with Python programming language Third Edition. The code development was done using MobaXterm, Nano, Notepad ++, Visual Studio Code, Qt Creator, Qt Designer, while the design and study of the three-dimensional parts of the paper were done using the Fusion 360 design software.

Keywords: Robotic system, Empathy development, Computer vision, Raspberry Pi, PyQt5, Open source software, 3D printing.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνουμε ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ.3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο «Σχεδίαση και υλοποίηση συστήματος ρομποτικής κατασκευής με υποστήριξη μηχανικής όρασης» καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη, αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχουμε χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και μόνο.

Ευχαριστίες

Αρχικά, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την αμέριστη υλική και ηθική συμπαράσταση τους κατά την διάρκεια των σπουδών μας, καθώς και την στήριξη τους σε κάθε μας επιλογή.

Στην συνέχεια, θα θέλαμε να προσθέσουμε ότι είμαστε ευγνώμονες στο κύριο Μηνά Δασυγένη, επιβλέπων καθηγητή της εργασίας μας, για την μεθοδική καθοδήγηση, την συνεργασία, την πολύτιμη βοήθεια του, καθώς και την εμπιστοσύνη που μας έδειξε για την ανάθεση της παρούσας διπλωματικής εργασίας.

Ευχαριστούμε τα μέλη του εργαστήριου Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας για την έμπρακτη στήριξη τους.

Ευχαριστούμε, τα μέλη της Ομάδα Ρομποτικής του Πανεπιστημίου Δυτικής Μακεδονίας, Hyperion Robotics, με τα οποία περάσαμε αξέχαστες στιγμές.

Τέλος, ευχαριστούμε τους φίλους μας που μας συμπαραστάθηκαν και μας βοήθησαν, ο καθένας με τον τρόπο του.

Κεφάλαιο 1ο – Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται λεπτομερώς το αντικείμενο της συγκεκριμένης εργασίας. Αρχικά αναφέρεται η σημερινή κατάσταση του συνεχώς εξελισσόμενου τεχνολογικού χώρου και διαπιστώνεται μέσα από ερευνά η ανάγκη ανάπτυξης της ενσυναίσθησης των ατόμων. Παρατίθενται οι λόγοι που οδήγησαν στην υλοποίηση του ρομπότ. Επίσης, παρουσιάζονται παρόμοιες υλοποιήσεις και πραγματοποιείται σύγκριση μεταξύ των συστημάτων.

1.1 Σημερινή Κατάσταση

Τα τελευταία χρόνια η τεχνολογική εξέλιξη στον χώρο των πληροφοριακών συστημάτων έχει γίνει αναπόσπαστο κομμάτι της εκπαιδευτικής διαδικασίας σε όλες τις βαθμίδες εκπαίδευσης. Με την βοήθεια τεχνολογικών εργαλείων κάθε εκπαιδευτική οντότητα κάνει την διαδικασία της μάθησης πιο ελκυστική, αποτελεσματική και άμεση. Άλλωστε οι ψηφιακές δεξιότητες που αποκτούν οι νέοι από πολύ μικρή ηλικία λόγω της ενασχόλησης τους με ανάλογες συσκευές, καθιστούν τις κάθε λογής νέες τεχνολογίες κομμάτι της καθημερινότητας τους και επομένως κατάλληλη οδός για την εκπαίδευσή τους σε διάφορους τομείς και επιστήμες.

Φανερό επιπλέον γίνεται, η αναγκαιότητα της ανάπτυξης της ικανότητας του ατόμου να κατανοεί τα συναισθήματα των ατόμων που συναναστρέφεται. Η ικανότητα αυτή αποτελεί σημαντικό κομμάτι στην ολόπλευρη ανάπτυξη των παιδιών [1]. Η ενσυναίσθηση τοποθετεί σωστά τα θεμέλια της υγιούς κοινωνικής συμπεριφοράς, εφόσον παρατηρείται ότι τα παιδιά της προσχολικής ηλικίας αναγνωρίζουν τον εαυτό τους ως ξεχωριστές οντότητες. Αποτελέσματα ερευνών οδηγούν στο συμπέρασμα ότι είναι αναγκαίο στο σχολείο να γίνεται προσπάθεια ανάπτυξης της γνωστικής και συναισθηματικής ενσυναίσθησης και της έκφρασης συναισθημάτων των παιδιών. Η ενσυναίσθηση παίζει καθοριστικό ρόλο στην συμπεριφορά και στην ανάπτυξη του χαρακτήρα τους και ως παιδιά, αλλά και ως μελλοντικοί ενήλικες.

Ο όρος «Διαταραχή του Αυτιστικού Φάσματος» (ΔΑΦ) [2] χρησιμοποιείται για την περιγραφή των ατόμων που αντιμετωπίζουν νευρο-αναπτυξιακές διαταραχές με κοινά χαρακτηριστικά, όπως τα ελλείμματα σε κοινωνικές-επικοινωνιακές δεξιότητες, τα περιορισμένα ενδιαφέροντα, καθώς και τα επαναληπτικά πρότυπα συμπεριφοράς.

Προκειμένου άτομα που χαρακτηρίζονται από νευρο-αναπτυξιακές διαταραχές να ενταχθούν στην κοινωνία μας είναι απαραίτητο να ξεκινήσει η εκπαίδευση και η ενημέρωση όλων των ατόμων από μικρή ηλικία. Συστήνεται από ειδικούς να συνεχίσει η ανάπτυξη παιχνιδιών και εργαλείων που θα βοηθούν τον εκπαιδευτικό να ενημερώσει και να εκπαιδεύσει τα παιδιά.

1.2 Σκοπός του έργου

Η άενη ανάπτυξη της τεχνολογίας και η εύκολη πρόσβαση στην ενημέρωση φέρνουν στο προσκήνιο ανάρμοστες συμπεριφορές προς άτομα που αντιμετωπίζουν διαταραχές στο φάσμα του αυτισμού. Υστέρα από αναζητήσεις για τρόπους εξάλειψης των φαινομένων αυτών παρατηρήθηκε ότι καθοριστικό ρολό παίζει η κατανόηση της συμπεριφοράς των άλλων ατόμων, καθώς και των κινήτρων τους. Στα πλαίσια αυτά γεννήθηκε η ιδέα για την δημιουργία ενός εργαλείου που θα βοηθήσει στην ερμηνεία του τρόπου συμπεριφοράς των ατόμων με ΔΑΦ.

Η κατασκευή ενός ρομποτικού συστήματος, το οποίο θα παράγει συναισθήματα και θα απευθύνεται σε παιδιά ηλικίας τεσσάρων με εννέα ετών θα προορίζεται για χρήση σε εκπαιδευτικές μονάδες, με στόχο την ανάπτυξη της ενσυναίσθησης τους. Κρίθηκε αναγκαία η μελέτη και ανάπτυξη σεναρίων τα οποία δίνουν στον εκπαιδευτικό την δυνατότητα να παρουσιάσει συμπεριφορές των ατόμων με ΔΑΦ, να συζητήσει και να προτείνει τις ενδεικνύμενες αντιδράσεις απέναντι σε αυτές τις συμπεριφορές. Ο εκπαιδευτικός είναι χρήσιμο να έχει την δυνατότητα να δημιουργήσει τα δικά του, σεναρία προσαρμοσμένα στις ανάγκες των μαθητών, προκειμένου να γίνουν πιο κατανοητές οι συμπεριφορές των ατόμων με ΔΑΦ.

Σκοπός του ρομπότ είναι να αποτελέσει ένα βοηθητικό εργαλείο, τόσο για την αποτελεσματικότερη ανάπτυξη της ενσυναίσθησης, όσο και για να εμπλουτίσει τον παραδοσιακό τρόπο διδασκαλίας με περισσότερες διαδραστικές μεθόδους.

1.3 Παρόμοιες Εργασίες

Μετά από μια ενδελεχή έρευνα σε διάφορους διαδικτυακούς τόπους εκπαιδευτικού και τεχνολογικού ενδιαφέροντος, παρουσιάζονται παρόμοιες υλοποιήσεις συστημάτων, που απευθύνονται στην ανάπτυξη της ενσυναίσθησης. Περιγράφονται τα χαρακτηριστικά των

υλοποιήσεων και οι περιπτώσεις που καλύπτουν. Στο τέλος, γίνεται σύγκριση όλων των εφαρμογών με την εφαρμογή που υλοποιήθηκε κατά την διάρκεια περάτωσης της εν λόγω έρευνας, καθώς παρατίθενται τα πλεονεκτήματα έναντι αυτών.

1.3.1 Moxie

Η Embodied είναι μία νεοφυής επιχείρηση (startup company) κοινωνικής ρομποτικής που δημιούργησε ένα νέο ρομπότ που ονομάζεται Moxie [3] (**Εικόνα 1**), ένα κοινωνικό σύντροφο για παιδιά ηλικίας 6 έως 9 ετών. Σύμφωνα με την Embodied, η Moxie έχει σχεδιαστεί για να συμβάλλει στην προώθηση της κοινωνικής, συναισθηματικής και γνωστικής ανάπτυξης μέσω της καθημερινής μάθησης με βάση το παιχνίδι. Η Moxie παρέχει τρόπους διδασκαλίας, που στηρίζονται στο παιχνίδι, με εβδομαδιαία εναλλαγή θεματολογίας και αποστολές, που περιεχόμενο τους έχουν την σχεδίαση, η οποία αποσκοπεί στην προώθηση της κοινωνικής, συναισθηματικής και γνωστικής μάθησης.

Περιλαμβάνει:

- Τακτικές ενημερώσεις νέου περιεχομένου και λειτουργιών.
- Απεριόριστη χρήση πόρων υπολογιστικού νέφους.
- Γονική εφαρμογή με καθημερινές πληροφορίες και συμβουλές από ειδικούς.
- Τα Moxie Mission Packs αποστέλλονται μέσω ταχυδρομείου κάθε μήνα.

Οι απαιτήσεις για την λειτουργία της Moxie είναι:

- Σύνδεση Wi-Fi.
- Smartphone (τουλάχιστον iOS 12 ή Android 6.0) για λήψη της εφαρμογής Moxie Parent.
- Ενεργή συνδρομή.



Εικόνα 1: Moxie εκπαιδευτικό ρομπότ (Embodied - <https://embodied.com/>)

1.3.2 QTrobot

Η LuxAI [4] είναι μια καινοτόμος εταιρεία που εξειδικεύεται σε ρομποτικές λύσεις για την εκπαίδευση και την υγειονομική περίθαλψη. Σκοπός της είναι η δημιουργία κοινωνικών βοηθητικών ρομπότ προσιτά σε όλους και φιλικά για τον χρήστη. Στόχος της είναι να επιτραπεί σε όλους τους εκπαιδευτικούς, τους θεραπευτές και τους παρόχους φροντίδας να χρησιμοποιούν προηγμένα ρομπότ για την καθημερινή τους εργασία.

Το QTrobot [5] (Εικόνα 2) είναι ένα εκφραστικό μικρό ανθρωποειδές ρομπότ, σχεδιασμένο και κατασκευασμένο με αφετηρία τις επιστημονικές προσεγγίσεις, για να βοηθήσει στη διδασκαλία νέων δεξιοτήτων σε παιδιά με αυτισμό και ειδικές εκπαιδευτικές ανάγκες. Το QTrobot υποστηρίζει τα παιδιά στην προσπάθειά τους να διδαχθούν τις άγνωστες έννοιες πάνω στις γνωστικές, κοινωνικές, επικοινωνιακές και συναισθηματικές δεξιότητες, καθώς είναι και αρωγός στο εγχείρημά τους να επωφεληθούν όσο το δυνατόν περισσότερο από τις εκπαιδευτικές συνεδρίες. Το QTrobot λειτουργεί ως εργαλείο για επαγγελματίες του αυτισμού και εκπαιδευτικούς για παιδιά με ειδικές εκπαιδευτικές ανάγκες, διευκολύνοντας την εργασία τους και αυξάνοντας την συνεργασία του παιδιού με τον ειδικό.

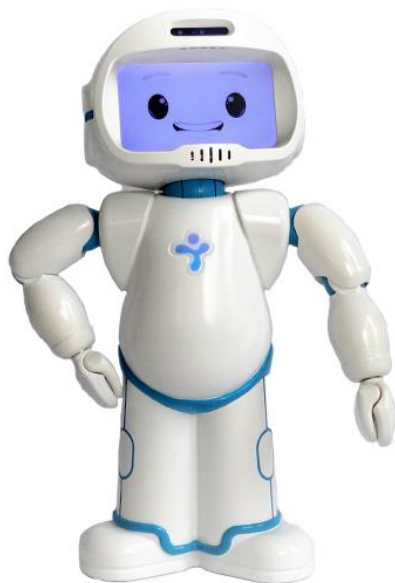
Το QTrobot συνοδεύεται από διάφορα εκπαιδευτικά προγράμματα που στοχεύουν σε διάφορες ηλικιακές ομάδες προκειμένου να αναπτυχθούν οι δεξιότητες των μαθητών και να γίνουν

πιο διευρυμένοι διανοητικά και πιο διαδραστικοί κοινωνικά, να αντιμετωπιστούν εν κατακλείδι ως ισότιμα και ικανά μέλη των εκάστοτε κοινωνιών.

Τα προγράμματα μαθημάτων έχουν σχεδιαστεί από ειδικούς του αυτισμού και διαθέτουν προδιαγραφές σύμφωνα με τις τεκμηριωμένες μεθόδους εκπαίδευσης. Το περιεχόμενο του QTrobot στην έκδοση που προορίζεται για παιδιά με αυτισμό είναι πλήρως προσαρμόσιμο με βάση τις ανάγκες του μαθητή. Οι επαγγελματίες του αυτισμού μπορούν επίσης να δημιουργήσουν τις δικές τους διδακτικές συνεδρίες χωρίς καμία γνώση πληροφορικής.

Περιλαμβάνει:

- Το ανθρωποειδές εκφραστικό ρομπότ QTrobot.
- Δύο τάμπλετ με προεγκατεστημένη εφαρμογή για εκπαιδευτή και για εφαρμογή μαθητή.
- Προ-προγραμματισμένα προσαρμόσιμα μαθήματα που αναπτύχθηκαν για μαθητές με αυτισμό.



Εικόνα 2: QTrobot εκπαιδευτικό ρομπότ (luxai - <https://luxai.com/>)

1.3.3 Σύγκριση

Αναλύοντας τα υφιστάμενα έως τώρα, συστήματα παρατηρούμε ότι και στις δύο περιπτώσεις τα ρομπότ διαθέτουν ανθρωπόμορφα χαρακτηριστικά με δυνατότητα έκφρασης

προσώπου και επικοινωνίας μέσω ομιλίας. Επίσης στο ένα σύστημα η ανάθεση λειτουργιών και χειρισμού του ρομπότ γίνεται μέσω μιας εφαρμογής κινητού ενώ στην δεύτερη διατίθενται δύο tablet με τα οποία αλληλεπιδρούν το παιδί και ο εκπαιδευτικός.

Το Moxie εξειδικεύεται κυρίως στην συντροφικότητα και στην βελτίωση της ικανότητας του παιδιού να συναναστρέφεται και επικοινωνεί με άλλα άτομα, ενώ το QTrobot εστιάζει στην βελτίωση δεξιοτήτων των παιδιών με ΔΑΦ που σχετίζονται με τη συγκέντρωση, την επικοινωνία και την κατάλληλη συμπεριφορά ανά περίπτωση.

Το σύστημα που παρουσιάζεται μέσω της συγκεκριμένης εργασίας στοχεύει στην αναγνώριση και κατανόηση των συναισθημάτων, που εκφράζονται από το ρομπότ, με σκοπό την ανάπτυξη της ενσυναίσθησης.

Το σύστημα που αναπτύχθηκε διαθέτει μια οθόνη αφής με την οποία πραγματοποιείται η διαχείριση του. Τα πλαστικά του μέρη είναι εκτυπωμένα σε τρισδιάστατο εκτυπωτή, συνεπώς δίνεται η δυνατότητα στους χρήστες να εκτυπώσουν κομμάτια που μπορεί να έχουν υποστεί φθορές ή να προσθέσουν διάφορα εξαρτήματα τα οποία θα συνοδεύονται από το αντίστοιχο σενάριο. Πέρα από τα βασικά προ εγκατεστημένα σενάρια που διαθέτει το ρομπότ, δίνεται η δυνατότητα ανάπτυξης και προσθήκης σεναρίων από τους ίδιους τους χρήστες μέσω της διασύνδεσης με πληροφοριακό σύστημα.

Μετά την ανάλυση των υλοποιήσεων, αδιαμφισβήτητα, διαπιστώνεται ότι το σύστημα, που αναπτύχθηκε στην συγκεκριμένη εργασία, δέχεται περαιτέρω ανάπτυξη προκειμένου να είναι ανταγωνιστικό.

1.4 Σύνοψη

Στο συγκεκριμένο κεφάλαιο συνοψίστηκαν οι προβληματισμοί που οδήγησαν στην εκπόνηση της συγκεκριμένης εργασίας. Ακολούθως, παρατέθηκαν οι προϋποθέσεις οι οποίες είναι αναγκαίο να καλύπτονται. Έγινε παρουσίαση και σύγκριση παρόμοιων υλοποιημένων συστημάτων. Στο επόμενο κεφάλαιο, εξετάζεται το θεωρητικό υπόβαθρο των λογισμικών και των τεχνολογιών, στις οποίες στηρίζεται το αντικείμενο μελέτης.

Κεφάλαιο 2ο – Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο αναφέρεται το θεωρητικό υπόβαθρο, όπου βασίστηκε η ανάπτυξη του ρομπότ, καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν. Αρχικά γίνεται ιστορική αναδρομή στις γλώσσες προγραμματισμού και στη συνέχεια αναφέρονται τα εργαλεία και υλικά που χρησιμοποιήθηκαν.

2.1 Λογισμικά - Εργαλεία

Στη ενότητα αυτή περιγράφονται τα λογισμικά που αξιοποιήθηκαν για την ανάπτυξη της παρούσα εργασίας. Χρησιμοποιήθηκαν λογισμικά που διαθέτουν δωρεάν άδειες χρήσης καθώς και λογισμικό, το οποίο διαθέτει δωρεάν ακαδημαϊκή άδεια χρήσης.

2.1.1 Python

Η Python [6] είναι διερμηνευόμενη, γενικού σκοπού και υψηλού επιπέδου γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού και υποστηρίζει τόσο τον διαδικαστικό όσο και τον αντικειμενοστραφή προγραμματιστικό. Είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) και υποστηρίζει μεγάλη συλλογή πακέτων. Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ στο ερευνητικό κέντρο Centrum Wiskunde & Informatica το 1989 και κυκλοφόρησε για πρώτη φορά το 1991.

Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ' ότι θα ήταν δυνατόν σε γλώσσες όπως C++ ή Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++.

Οι διερμηνευτές της Python [7] (**Εικόνα 3**) είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα

συστημάτων. Η Python αναπτύσσεται ως ανοιχτό λογισμικό και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation.



Εικόνα 3: Παράδειγμα κώδικα σε Python με το λογότυπο της γλώσσας

2.1.2 OpenCV Python

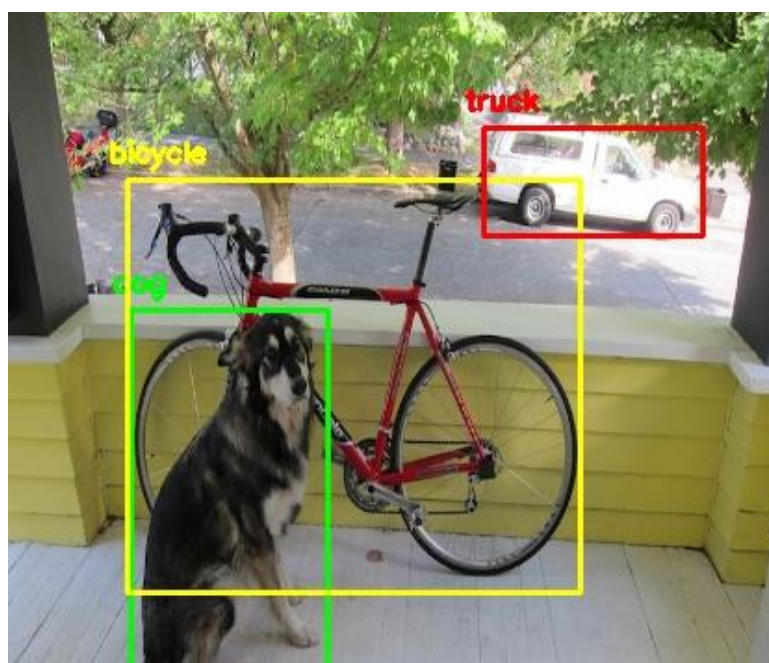
Το OpenCV [8] ξεκίνησε στην Intel το 1999 από τον Γκάρι Μπράντσκι και κυκλοφόρησε το 2000. Ο Βαντίμ Πισαρέβσκι εντάχθηκε στη διαχείριση της ρωσικής ομάδας λογισμικού OpenCV της Intel. Αργότερα, η ανάπτυξη συνέχισε υπό την υποστήριξη του Γουίλλου Γκαραζ με τους Γκάρι Μπράντσκι και Βαντίμ Πισαρέβσκι να ηγούνται του έργου. Το OpenCV υποστηρίζει πλέον ένα πλήθος αλγορίθμων που σχετίζονται με τον τομέα της υπολογιστικής όρασης και τον κλάδο της μηχανικής εκμάθησης.

Το OpenCV υποστηρίζει μια μεγάλη ποικιλία γλωσσών προγραμματισμού όπως C ++, Python, Java κ.λπ. και διατίθεται σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των Windows, Linux, OS X, Android και iOS. Οι διεπαφές για λειτουργίες υψηλής ταχύτητας της γραφικής μονάδα επεξεργασίας (Graphical Process Unit - GPU) οι οποίες βασίζονται στον υπολογισμό ενοποιημένης αρχιτεκτονικής συσκευών (Compute Unified Device Architecture - CUDA) και της ανοιχτής υπολογιστικής γλώσσας (Open Computing Language - OpenCL) βρίσκονται ακόμα σε εξέλιξη.

Το OpenCV-Python (**Εικόνα 4**) είναι μια βιβλιοθήκη διεπαφών Python που έχει σχεδιαστεί για την επίλυση προβλημάτων όρασης υπολογιστή. Το OpenCV-Python είναι το API Python για

OpenCV, συνδυάζοντας τις καλύτερες ποιότητες του OpenCV C++ API και της γλώσσας Python. Το OpenCV-Python χρησιμοποιεί το Numpy, το οποίο είναι μια εξαιρετικά βελτιστοποιημένη βιβλιοθήκη για αριθμητικές λειτουργίες με σύνταξη τύπου MATLAB. Όλες οι δομές πίνακα OpenCV μετατρέπονται από και σε συστοιχίες Numpy. Αυτό διευκολύνει επίσης την ενσωμάτωση με άλλες βιβλιοθήκες που χρησιμοποιούν το Numpy, όπως το SciPy και το Matplotlib.

Το εικονικό περιβάλλον (virtual environment) [9] του OpenCV είναι ένα συνεργατικό απομονωμένο περιβάλλον εκτέλεσης που επιτρέπει στους χρήστες και τις εφαρμογές της Python να εγκαταστήσουν και να αναβαθμίσουν πακέτα διανομής Python χωρίς να παρεμβαίνουν στη συμπεριφορά άλλων εφαρμογών Python που εκτελούνται στο ίδιο σύστημα.



Εικόνα 4: OpenCV αναγνώριση αντικείμενων

2.1.3 PyQt v5

Το PyQt πέμπτης έκδοσης [10] είναι ένα σύνολο διεπαφών της γλώσσας προγραμματισμού Python των εκδόσεων 2 και 3, στο πλαίσιο της εφαρμογής Qt. Εκτελείται στις πλατφόρμες Windows, macOS, Linux, iOS και Android που υποστηρίζετε και η Qt. Το PyQt (Εικόνα 5) πέμπτης έκδοσης υποστηρίζει την πέμπτη έκδοση της εφαρμογής Qt. Η τέταρτη έκδοση της PyQt δεν υποστηρίζεται πλέον, αντίθετα συνίσταται η πέμπτη έκδοση για κάθε νέα ανάπτυξη. Υπάρχουν

δύο τύπων άδειες για τις υποστηριζόμενες πλατφόρμες, η πρώτη είναι η GNU GPL v3, η οποία είναι μια γενική άδεια δημόσιας χρήσης ελεύθερου λογισμικού, και η δεύτερη είναι η εμπορική άδεια Riverbank. Αντίθετα με το Qt, το PyQt δεν είναι διαθέσιμο στη Lesser General Public License (LGP), η οποία είναι η μικρότερη γενική άδεια ελεύθερου λογισμικού που εκδίδεται από το ίδρυμα ελεύθερου λογισμικού. Το PyQt δεν συμπεριλαμβάνεται στο αντίγραφο του Qt.

Χαρακτηριστικό του PyQt είναι ότι συνδυάζει πολλαπλές εφαρμογές πλατφορμών του Qt, C++ και την γλώσσα προγραμματισμού Python. Το Qt εκτός από τα εργαλεία κατασκευής γραφικού περιβάλλοντος χρήστη (Graphical User Interface) διαθέτει και άλλες λειτουργίες. Μερικές από αυτές είναι η υποστήριξη υποδοχών δικτύου (network sockets), νημάτων (threads), του προτύπου Unicode, κανόνων έκφρασης, βάσεων δεδομένων SQL, αρχείων εικόνων SVG, της διεπαφής προγραμματισμού εφαρμογών για την απόδοση δισδιάστατων και τρισδιάστατων γραφικών OpenGL, της γλώσσας XML η οποία περιέχει ένα σύνολο κανόνων για ηλεκτρονική κωδικοποίηση κειμένων, ενός πλήρους λειτουργικού προγράμματος περιήγησης ιστού, ενός συστήματος βοήθειας και ενός πλαισίου πολυμέσων. Τέλος υπάρχει στην διάθεση του χρήστη μια πλούσια συλλογή γραφικών στοιχείων GUI.

Επίσης, βασικό πλεονέκτημα των κλάσεων Qt αποτελεί ο μηχανισμός σήματος/υποδοχής που χρησιμοποιούν για την ασφαλή επικοινωνία μεταξύ αντικειμένων, αλλά με χαλαρή διασύνδεση, δημιουργώντας εύκολα επαναχρησιμοποιήσιμα στοιχεία λογισμικού.

Για την καλύτερη και ταχύτερη δημιουργία και συντήρηση γραφικού περιβάλλοντος, η Qt διαθέτει επίσης το Qt Designer, ένα πρόγραμμα σχεδιασμού διεπαφής χρήστη. Μέσα από το συγκεκριμένο πρόγραμμα, δίνεται η δυνατότητα δημιουργίας κώδικα στην γλώσσα Python μέσα από ένα γραφικό περιβάλλον, στο οποίο οι χρήστες έχουν πολυάριθμες επιλογές στοιχείων γραφικού περιβάλλοντος.



Εικόνα 5: Λογότυπο PyQt

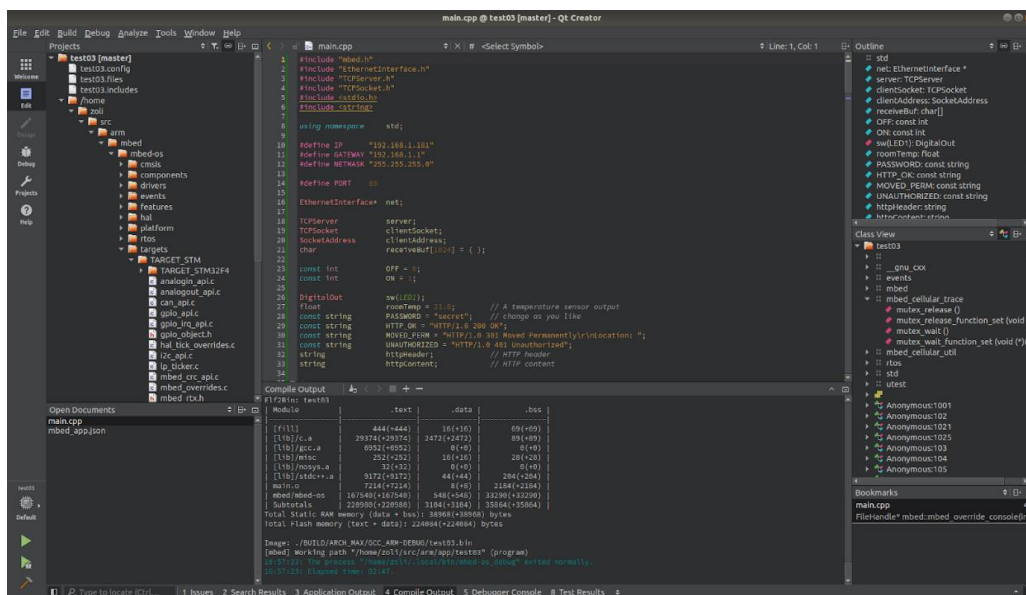
2.1.4 Qt Creator

Το Qt Creator [11] είναι ένα λογισμικό πολλαπλών πλατφόρμων, υποστηρίζεται από C++, JavaScript και QML. Το ολοκληρωμένο περιβάλλον ανάπτυξης που διαθέτει, διευκολύνει την ανάπτυξη εφαρμογών γραφικού περιβάλλοντος χρήστη (Graphical User Interface - GUI). Εμπεριέχει ένα οπτικό πρόγραμμα εντοπισμού σφαλμάτων και μια ολοκληρωμένη διάταξη γραφικού περιβάλλοντος χρήστη (GUI) “Αυτό που βλέπετε, είναι αυτό που παίρνετε” (What You See Is What You Get – WYSIWYG) . Η συγκεκριμένη διάταξη αναφέρεται σε ένα σύστημα, στο οποίο το λογισμικό επεξεργασίας επιτρέπει την επεξεργασία περιεχομένου σε μορφή που μοιάζει με την εμφάνιση της όταν εκτυπώνεται ή εμφανίζεται σαν τελικό προϊόν.

Χαρακτηριστικά του λογισμικού (**Εικόνα 6**) είναι η επισήμανση σύνταξης και η αυτόματη συμπλήρωση. Το Qt χρησιμοποιεί τον μεταγλωττιστή C++ από την συλλογή μεταγλωττιστών του GNU (GCC) σε Linux και FreeBSD. Στα Windows, χρησιμοποιεί το MinGW ή το MSVC, το πρώτο είναι ένα περιβάλλον ανάπτυξης δωρεάν λογισμικού και ανοιχτού κώδικα για την δημιουργία εφαρμογών Microsoft Windows, ενώ το δεύτερο είναι ένα προϊόν ολοκληρωμένου περιβάλλοντος ανάπτυξης από τη Microsoft για τις γλώσσες προγραμματισμού C, C++ και C++/CLI. Πέραν των προαναφερθέντων μεταγλωττιστών, οι οποίοι είναι προ εγκατεστημένοι, μπορεί να χρησιμοποιηθεί και το Microsoft Console Debugger, όταν μεταγλωττίζεται από πηγαίο κώδικα σε εκτελέσιμο κώδικα. Επίσης υποστηρίζεται το Clang, ο οποίος είναι ένας μεταγλωττιστής για τις γλώσσες προγραμματισμού C, C++ και πολλές άλλες.

Ο creator ξεκίνησε σαν project Workbench και GreenHouse, το 2007 πήρε το όνομα Qt Creator. Πραγματοποίησε την πρώτη του εμφάνιση στο τέλος της τέταρτης έκδοσης Qt, αρχίζοντας με την κυκλοφορία του Qt Creator πρώτης έκδοσης τον Μάρτιο του 2009, στην συνέχεια ομαδοποιήθηκε με το Qt 4.5 στο κιτ ανάπτυξης λογισμικού (SDK) 2009.3. Μέχρι τότε η αυτόνομη εφαρμογή Qt Designer ήταν το εργαλείο διάταξης widget του προγραμματιστή. Τα έργα που μπορεί να διαχειριστεί το Qt Creator είναι ποικίλα, ενδεικτικά μπορεί να διαχειριστεί αρχεία τύπου .pro, CMake, Autotools και άλλα. Ένα αρχείο έργου μπορεί να διαθέτει πληροφορίες, όπως τα αρχεία τα οποία συμπεριλαμβάνονται στο έργο, προσαρμοσμένα βήματα κατασκευής και ρυθμίσεις για την εκτέλεση των εφαρμογών.

Το Qt Creator περιλαμβάνει έναν συντάκτη κώδικα και ενσωματώνει το Qt Designer για τον σχεδιασμό και την δημιουργία γραφικών διεπαφών χρήστη (GUI). Το λογισμικό έχει την δυνατότητα επισήμανσης σύνταξης για διάφορες γλώσσες, ενώ αναλύει κώδικα με αποτέλεσμα να εξάγεται σε γλώσσες C++ και σε μια γλώσσα διεπαφής χρήστη την QML. Επιπλέον, παρέχει βοήθεια στο συγκεκριμένο σημείο (context-sensitive) και σημασιολογική πλοήγηση.



Εικόνα 6: Περιβάλλον Qt Creator

2.1.5 GCompris

Το GCompris [12] είναι λογισμικό υψηλής εκπαιδευτικής ποιότητας και αποτελείται από δραστηριότητες για παιδιά ηλικίας 2 έως 10 ετών. Οι δραστηριότητες που διαθέτει αποτελούνται κυρίως από παιχνίδια, παρόλα αυτά διατηρούν τον εκπαιδευτικό τους χαρακτήρα.

Παρακάτω παρατίθεται μια λίστα των κατηγοριών με μερικά παραδείγματα:

- ανακάλυψη του υπολογιστή: πληκτρολόγιο, ποντίκι, οθόνη αφής
- ανάγνωση: γράμματα, λέξεις, εξάσκηση ανάγνωσης, πληκτρολόγηση κειμένου
- αριθμητική: αριθμοί, πράξεις, πίνακας μνήμης, αρίθμηση, πίνακας διπλής καταχώρησης
- επιστήμη: λειτουργία διώρυγας, ο κύκλος του νερού, υποβρύχιο, ηλεκτρικά κυκλώματα
- γεωγραφία: χώρες, περιοχές, πολιτισμός

- παιχνίδια: σκάκι, μνημονικές τεχνικές, σύνδεσε 4, oware, sudoku
- άλλα: χρώματα, σχήματα, Μπράιγ, μάθε την ώρα

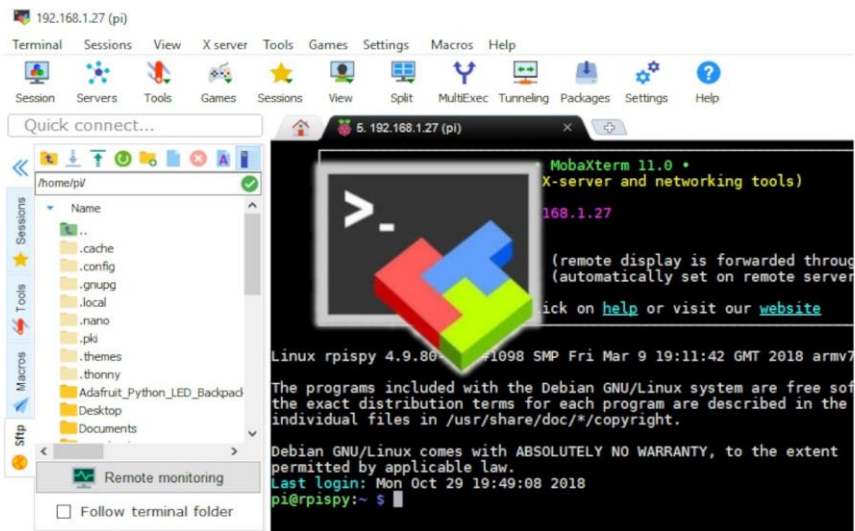
Αυτή τη στιγμή το GCompris (Εικόνα 7) προσφέρει περισσότερες από 100 δραστηριότητες, ενώ αρκετές βρίσκονται στο στάδιο της ανάπτυξης. Το GCompris είναι ελεύθερο λογισμικό, που σημαίνει ότι μπορεί ο οποιοσδήποτε να το προσαρμόσει στις δικές του ανάγκες και να το βελτιώσει.



Εικόνα 7: Στιγμιότυπο αρχικής οθόνης GCompris

2.1.6 MobaXterm

Το MobaXterm [13] είναι μια εφαρμογή για λειτουργικά συστήματα Windows, η οποία προσφέρει πολλαπλές λειτουργίες για προγραμματιστές και διαχειριστές πληροφοριακών συστημάτων για τον χειρισμό απομακρυσμένου ελέγχου. Υποστηρίζει διάφορα πρωτόκολλα, όπως SSH, VNC, FTP. Πιο συγκεκριμένα, υπάρχει η επιλογή χρήσης του ασφαλούς πρωτοκόλλου SSH (Εικόνα 8) με παράλληλη απεικόνιση γραφικού προγράμματος περιήγησης του πρωτοκόλλου SFTP. Επίσης, το τερματικό που διαθέτει η εφαρμογή επισημαίνει τη σύνταξη και χρωματίζει τις λέξεις κλειδιά, ακόμα και αν δεν υποστηρίζεται από τον διακομιστή.



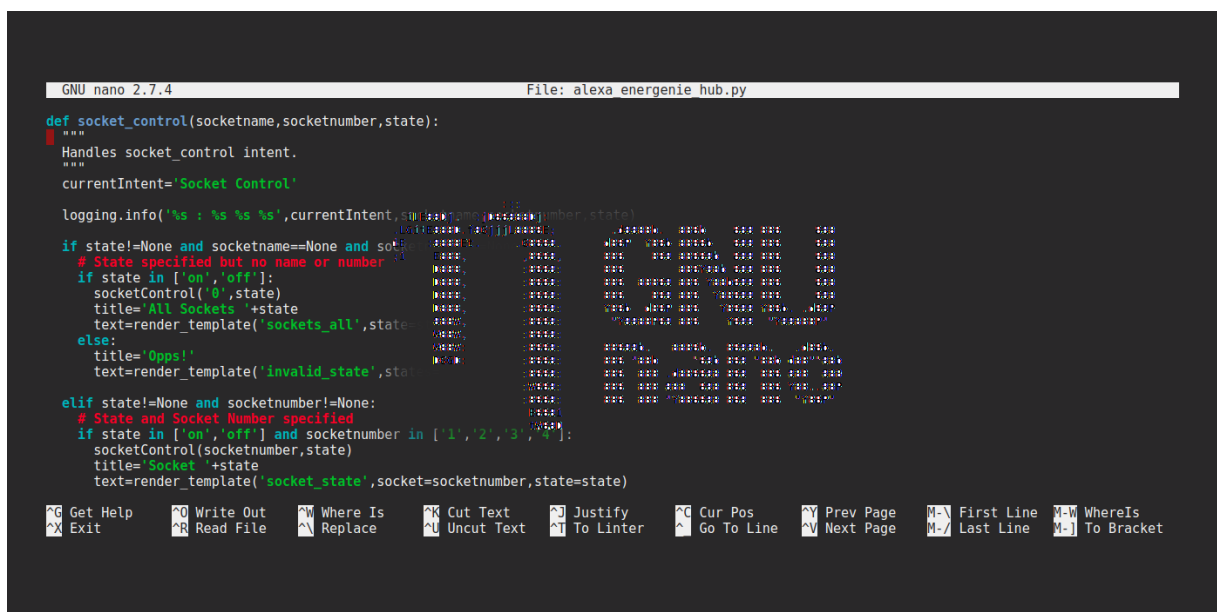
Εικόνα 8: Παράδειγμα σύνδεσης SSH σε MobaXterm με το λογότυπο του προγράμματος

2.1.7 Nano

Το nano [14] είναι ένας κειμενογράφος, ο οποίος βασίζεται στην βιβλιοθήκη curses και χρησιμοποιείται σε συστήματα Unix. Είναι βασισμένο σε ένα παλαιότερο εργαλείο ανάπτυξης κείμενου ονόματι Pico. Το nano δημιουργήθηκε το 1999 με την ονομασία TIP από τον Κρίς Αλλέγκρετα. Στόχος του ήταν να δημιουργήσει μια ελεύθερη έκδοση του λογισμικού Pico, καθώς δεν υπήρχε κάτι αντίστοιχο σε ελεύθερη μορφή. Το όνομα του λογισμικού Nano επισημοποιήθηκε στις 10 Ιανουαρίου 2000 για να μην υπάρχουν παρεξηγήσεις με το εργαλείο Tip. Η επιλογή του ονόματός του προέκυψε από το διεθνές σύστημα μονάδων (SI), στο οποίο το nano είναι μια μονάδα μέτρησης χίλιες φορές μεγαλύτερη από την pico. Το Φεβρουάριο του 2001, το nano έγινε μέλος του λειτουργικού συστήματος GNU.

Ο κειμενογράφος nano (**Εικόνα 9**) εμπλούτισε τις δυνατότητές του σε σχέση με τον Pico, καθώς προσέθεσε χρωματισμό κειμένου, αναζήτηση και αντικατάσταση κανονικών εκφράσεων, ομαλή κύλιση και πολλαπλές περιοχές κειμένων. Ο Κρίς Αλλέγκρετα παρέδωσε την συντήρηση του nano στις 11 Αυγούστου του 2003 στον Ντέιβιντ Λόρενς Ράμσεϊ, ο οποίος ανακοίνωσε ότι παραιτείται από την συντήρησή του στις 20 Δεκεμβρίου του 2007. Τα στοιχεία που βρίσκονται

υπό ανάπτυξη για τη σειρά έκδοσης 2.1 είναι η υποστήριξη ρυθμιζόμενων συντομεύσεων πλήκτρων και η λειτουργία undo.



```
GNU nano 2.7.4 File: alexa_energenie_hub.py
def socket_control(socketname,socketnumber,state):
    """
    Handles socket_control intent.
    """
    currentIntent='Socket Control'

    logging.info('%s : %s %s %s',currentIntent,socketname,socketnumber,state)

    if state!=None and socketname==None and socketnumber!=None:
        # State specified but no name or number
        if state in ['on','off']:
            socketControl('0',state)
            title= 'All Sockets' +state
            text=render_template('sockets_all',state)
        else:
            title='Oops!'
            text=render_template('invalid_state',state)
    elif state!=None and socketnumber!=None:
        # State and Socket Number specified
        if state in ['on','off'] and socketnumber in [1,2,3,4]:
            socketControl(socketnumber,state)
            title= 'Socket' +state
            text=render_template('socket_state',socket=socketnumber,state=state)
```

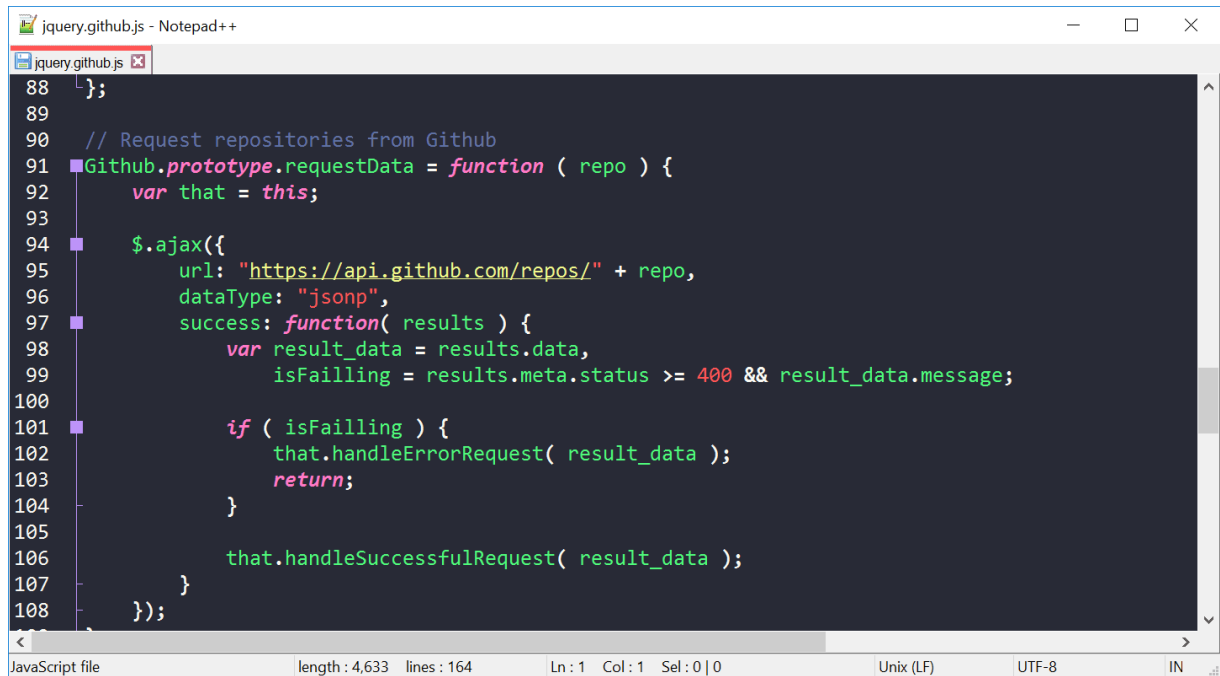
Εικόνα 9: Παράδειγμα του κειμενογράφου Nano

2.1.8 Notepad++

Το Notepad++ [15] είναι ένα πρόγραμμα επεξεργασίας κειμένου και πηγαίου κώδικα για τα λειτουργικά συστήματα Windows. Τον Σεπτέμβριο του 2003, ο Ντον Χο το δημιούργησε και κυκλοφόρησε για πρώτη φορά. Μερικά από τα βασικά του χαρακτηριστικά που τον κάνουν να ξεχωρίζει είναι η επισήμανση σύνταξης, η αναδίπλωση κώδικα, η περιορισμένη αυτόματη συμπλήρωση και η δέσμη ενεργειών και σήμανσης. Στην βασική έκδοση δεν διαθέτει έξυπνη ολοκλήρωση κώδικα ή έλεγχο σύνταξης.

Ο χρήστης έχει τη δυνατότητα να επεξεργάζεται ξεχωριστά τα αρχεία σε πολλαπλές καρτέλες. Επίσης, υποστηρίζει την επιλογή εργασίας σε πολλά αρχεία σε ένα παράθυρο. Το όνομα του είναι εμπνευσμένο από τον τελεστή αύξησης στην γλώσσα προγραμματισμού C. Το Notepad++ (Εικόνα 10) αποτελεί ένα δωρεάν λογισμικό το οποίο ξεκίνησε την διανομή του από το SourceForge.net. Από εκεί έγινε λήψη πάνω από είκοσι οκτώ εκατομμύρια φορές σε ένα χρόνο, με αποτέλεσμα να κερδίσει δύο φορές το βραβείο Επιλογής Κοινότητας SourceForge για το

καλύτερο εργαλείο προγραμματιστών. Έπειτα, το λογισμικό φιλοξενήθηκε στο TuxFamily, Από το 2010 φιλοξενείται στο GitHub. Το Notepad++ χρησιμοποιεί το επεξεργαστικό στοιχείο Scintilla, το οποίο αποτελεί μια δωρεάν βιβλιοθήκη ανοιχτού κώδικα.



```
88  });
89
90  // Request repositories from Github
91  Github.prototype.requestData = function ( repo ) {
92      var that = this;
93
94      $.ajax({
95          url: "https://api.github.com/repos/" + repo,
96          dataType: "json",
97          success: function( results ) {
98              var result_data = results.data,
99                  isFailing = results.meta.status >= 400 && result_data.message;
100
101              if ( isFailing ) {
102                  that.handleErrorRequest( result_data );
103                  return;
104              }
105
106              that.handleSuccessfulRequest( result_data );
107          }
108      });

```

JavaScript file length : 4,633 lines : 164 Ln : 1 Col : 1 Sel : 0 | 0 Unix (LF) UTF-8 IN

Εικόνα 10: Παράδειγμα του κειμενογράφου Notepad++

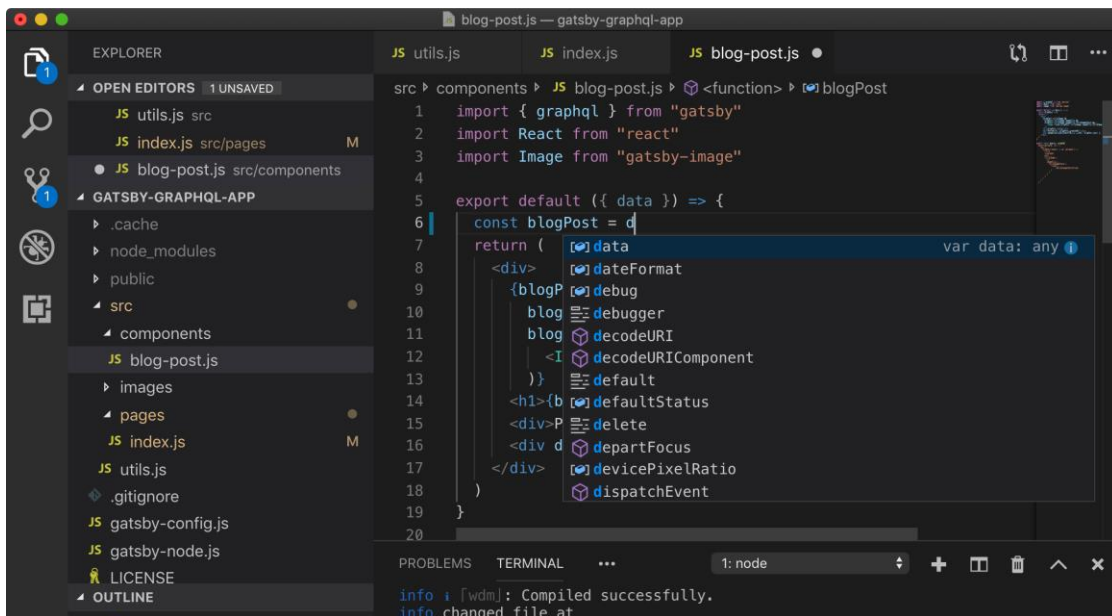
2.1.9 Visual Studio Code

Το Visual Studio Code [16] είναι ένας δωρεάν συντάκτης πηγαίου κώδικα που δημιουργήθηκε από την Microsoft για λειτουργικά περιβάλλοντα Windows, Linux, macOS. Διαθέτει λειτουργίες για εντοπισμό σφαλμάτων, επισήμανση σύνταξης, έξυπνη ολοκλήρωση κώδικα, αποσπάσματα, αναδιαμόρφωση κώδικα και περιέχει ενσωματωμένο Git. Προσαρμόζεται στις προτιμήσεις του κάθε χρήστη έχοντας την επιλογή αλλαγής θέματος και αλλαγής των συντομεύσεων του πληκτρολογίου. Υπάρχει και η δυνατότητα εγκατάστασης επεκτάσεων που του προσθέτουν επιπλέον λειτουργικότητα.

Ο πηγαίος κώδικας του Visual Studio Code (**Εικόνα 11**) έχει βάση το δωρεάν λογισμικό ανοιχτού κώδικα της Microsoft. Το πρόγραμμα κυκλοφορεί με την ανεκτική άδεια ελεύθερου λογισμικού Expat που προέρχεται από το ινστιτούτο τεχνολογίας της Μασαχουσέτης (MIT).

Επίσης, τα μεταγλωττισμένα δυαδικά αρχεία διατίθενται δωρεάν για οποιονδήποτε χρήστη. Επίσημη ανακοίνωση του λογισμικού έγινε στις 29 Απριλίου 2015 στο συνέδριο Build από την Microsoft. Έπειτα κυκλοφόρησε και μια έκδοση για προεπισκόπηση του λογισμικού. Στις 18 Νοεμβρίου του 2015 κυκλοφόρησε με ανεκτική άδεια ελεύθερου λογισμικού Expat και ο πηγαίος του κώδικας δημοσιεύτηκε στο GitHub, ενώ ανακοινώθηκε και η υποστήριξη επεκτάσεων από το λογισμικό.

Το Visual Studio Code υποστηρίζει αρκετές γλώσσες προγραμματισμού, όπως είναι η Java, η JavaScript, η Go, η Node.js και η C++. Βασίζεται σε ένα ανοιχτού λογισμικού πλαίσιο που αναπτύχθηκε από το GitHub και ονομάζεται Electron. Το Electron χρησιμοποιείται για την ανάπτυξη εφαρμογών ιστού στην γλώσσα προγραμματισμού Node.js που εκτελείται στην μηχανή αναζήτησης Blink.

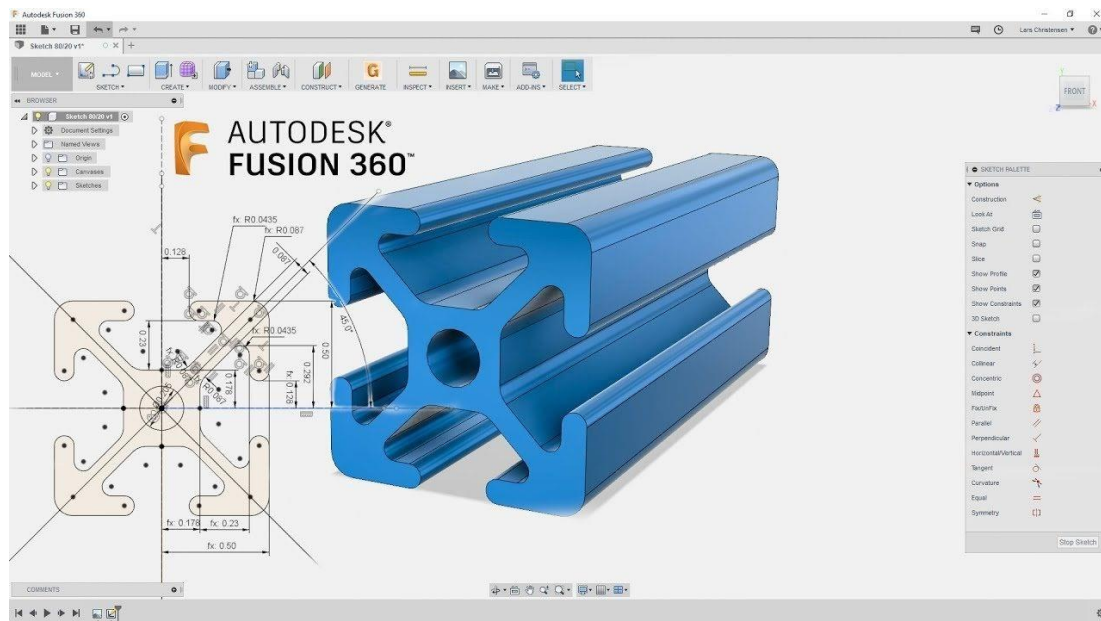


Εικόνα 11: Περιβάλλον Visual Studio Code

2.1.10 Fusion 360

Το Fusion 360 [17] είναι ένα εργαλείο σχεδίασης τρισδιάστατων αντικειμένων με τη βοήθεια του υπολογιστή (3D CAD) και ανήκει στην Autodesk. Είναι διαθέσιμο σε αρκετές πλατφόρμες συμπεριλαμβανομένων των Windows, Mac και σε πρόγραμμα περιήγησης. Άλλα

παρόμοια προγράμματα τρισδιάστατης σχεδίασης είναι το Solidworks και Siemens NX. Το Fusion 360 επιτρέπει στον χρήστη να αποκτήσει σφαιρική εικόνα για μια ολοκληρωμένη σειρά εργαλείων μοντελοποίησης. Προσφέρει την επιλογή δημιουργίας και επεξεργασίας σκίτσων. Χαρακτηριστικά του Fusion 360 (Εικόνα 12) είναι η άμεση μοντελοποίηση, επεξεργασία και επιδιόρθωση σχεδίων. Το πρόγραμμα επιτρέπει την παραμετρική μοντελοποίηση, δηλαδή προσφέρει λειτουργίες βάση ιστορικού, όπως extrude, revolve, loft, sweep και άλλα, τα οποία ενημερώνονται με αλλαγές στο σχεδιασμό. Με το Fusion 360, υπάρχει η δυνατότητα επεξεργασίας και επιδιόρθωσης εισαγόμενων σαρώσεων ή μοντέλων πλέγματος, συμπεριλαμβανομένων αρχείων STL και OBJ.



Εικόνα 12: Περιβάλλον Fusion 360

2.1.11 Prusa i3 mk3

Ο Prusa i3 mk3 [18] είναι ένας τρισδιάστατος εκτυπωτής μοντελοποίησης εναπόθεσης (FDM) ανοιχτού κώδικα (open source) και κατασκευάστηκε από την Τσέχικη εταιρεία Prusa Research. Αποτελεί μέρος του έργου RepRap, το οποίο ξεκίνησε στην Αγγλία το 2005 για την ανάπτυξη ενός τρισδιάστατου εκτυπωτή χαμηλού κόστους που μπορεί να εκτυπώσει τα

περισσότερα δομικά του μέρη. Το Prusa i3 mk3 (**Εικόνα 13**) κυκλοφόρησε τον Σεπτέμβριο του 2017 με σημαντικές βελτιώσεις συγκριτικά με τα προηγούμενα μοντέλα. Λόγω του ότι ο εκτυπωτής είναι ανοιχτού κώδικα, υπάρχουν αρκετές παραλλαγές που δημοσιεύονται από εταιρείες και ιδιώτες παγκοσμίως. Ο τρισδιάστατος εκτυπωτής Prusa i3 είναι το τρίτο μοντέλο του Τζόσεφ Προύσα, ένας από τους βασικούς προγραμματιστές του έργου RepRap.

Ο Prusa i3 mk3 διαθέτει την νέα mk52 μαγνητική πλάκα εκτύπωσης η οποία διατηρεί ένα αντικαταστάσιμο φύλλο χάλυβα ελατηρίου από κράμα με επιφάνεια πολυαιθυλενιμίνης (PEI). Υπάρχουν δύο επιλογές μαγνητικής πλάκας εκτύπωσης, η πρώτη είναι το λείο φύλλο πολυαιθυλενιμίνης (PEI) και το δεύτερο έχει υφή αδρή με επίστρωση σκόνης πολυαιθυλενιμίνης (PEI). Υποστηρίζει γρήγορη και αθόρυβη εκτύπωση, με ταχύτητα εκτύπωσης μεγαλύτερη από διακόσια χιλιοστά το δευτερόλεπτο (200 mm/s). Επίσης, έχει την προσθήκη ανίχνευσης απώλειας ρεύματος, έτσι ώστε να διακόψει την θέρμανση της πλάκας εκτύπωσης (heatbed) και του εξωθητή (extruder) και με την επαναφορά του ρεύματος να συνεχίσει την εκτύπωση. Επιπλέον, διαθέτει αισθητήρα νήματος, ο οποίος ελέγχει την ύπαρξη του υλικού εκτύπωσης προκειμένου να μην συνεχίσει την εκτύπωση στην περίπτωση που δεν υπάρχει πλέον υλικό. Ακόμα, ο Prusa i3 mk3 έχει ενσωματώσει δύο θερμίστορες, τα οποία μετράνε την θερμοκρασία στα ηλεκτρονικά, αποτρέποντας σφάλματα ελάχιστης θερμοκρασίας για τιμές κάτω των δεκαπέντε βαθμών κελσίου. Τέλος, ο τρισδιάστατος εκτυπωτής Prusa i3 mk3 χρησιμοποιεί το ακροφύσιο E3D V6, το οποίο θεωρείται κορυφαίο στην αγορά για τις υψηλές θερμοκρασίες του.



Εικόνα 13: Εκτυπωτής Prusa i3 mk3

2.2 Υλικός εξοπλισμός

Στη ενότητα αυτή περιγράφονται τα υλικά και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του ρομπότ.

2.2.1 Raspberry Pi

Το Raspberry Pi [19] είναι μια σειρά μικρών υπολογιστών μονής πλακέτας που αναπτύσσονται στο Ηνωμένο Βασίλειο από το Ίδρυμα Raspberry Pi. Νωρίς το έργο Raspberry Pi έστρεψε την προώθηση της διδασκαλίας βασικής επιστήμης υπολογιστών σε σχολεία και σε αναπτυσσόμενες χώρες. Αργότερα, το αρχικό μοντέλο έγινε περισσότερο δημοφιλές από το αναμενόμενο, πουλώντας έξω από την αγορά-στόχο του για χρήσεις όπως η ρομποτική.



Μετά την κυκλοφορία του δεύτερου τύπου, το Ίδρυμα Raspberry Pi ίδρυσε μια νέα οντότητα, με την ονομασία Raspberry Pi Trading. Το Ίδρυμα επαναπροσδιορίστηκε ως εκπαιδευτικό φιλανθρωπικό ίδρυμα για την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών σε σχολεία και αναπτυσσόμενες χώρες.


Τα μοντέλα Raspberry Pi είναι οι Βρετανικοί υπολογιστές με τις καλύτερες πωλήσεις. Από τον Δεκέμβριο του 2019, έχουν πωληθεί περισσότερα από τριάντα εκατομμύρια πλακέτες. Τα περισσότερα Pis κατασκευάζονται σε εργοστάσιο της Sony στο Pencoed της Ουαλίας, ενώ άλλα κατασκευάζονται στην Κίνα και την Ιαπωνία.

Η οθόνη αφής επτά ιντσών της Raspberry Pi [20] δίνει στους χρήστες την δυνατότητα να δημιουργήσουν ολοκληρωμένα έργα τύπου “όλα σε ένα” (all-in-one), όπως τάμπλετ (tablet), συστήματα ψυχαγωγίας και ενσωματωμένα έργα. Ένα προτεινόμενο έργο, από την ίδια την εταιρία είναι η δημιουργία ενός συστήματος Internet of Things (IoT), στο οποίο με την βοήθεια της γλώσσας προγραμματισμού Python, ο χρήστης μπορεί να αναπτύξει ένα σενάριο όπου θα υπάρχει αλληλεπίδραση μέσω της οθόνης. Τέλος, υπάρχει μια σειρά εκπαιδευτικών λογισμικών και προγραμμάτων τα οποία είναι διαθέσιμα στο Raspberry Pi και μπορούν να διευκολύνουν τη μάθηση, με τη διαδικασία αυτή να επιτυγχάνεται με άγγιγμα.

Η Raspberry Pi [21] κάμερα προσφέρει τη δυνατότητα λήψης βίντεο υψηλής ευκρίνειας, καθώς και λήψη φωτογραφιών εξίσου υψηλής ευκρίνειας. Είναι εύκολη στην χρήση, το οποίο είναι αρκετά βοηθητικό για αρχάριους χρήστες, αλλά μπορεί να προσφέρει πολλά και σε

προχωρημένους χρήστες. Μερικά παραδείγματα εφαρμογής της κάμερας είναι η χρονική καθυστέρηση, η αργή κίνηση αλλά και τα έξυπνα βίντεο, στα οποία γίνεται μια επεξεργασία της λήψης και σαν αποτέλεσμα υπάρχει ένα εμπλουτισμένο βίντεο.


Συσκευή	Τεχνικά Χαρακτηριστικά
 <p data-bbox="383 936 667 968">Εικόνα 14: Raspberry Pi 4</p>	<ul style="list-style-type: none"> • Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz • SDRAM 4 GB LPDDR4-3200 • 2,4 GHz και 5,0 GHz IEEE 802.11ac ασύρματο, Bluetooth 5.0, BLE • Gigabit Ethernet • 2 θύρες USB 3.0. 2 θύρες USB 2.0. • 2 × θύρες micro-HDMI (υποστηρίζονται έως και 4k@60) • 5V DC μέσω υποδοχής USB-C • Θερμοκρασία λειτουργίας: 0 - 50 βαθμοί C περιβάλλοντος
 <p data-bbox="383 1539 667 1570">Εικόνα 15: Οθόνη αφής 7"</p>	<ul style="list-style-type: none"> • Οθόνη αφής 7 ιντσών • Ανάλυση οθόνης 800 x 480 pixels • Σύνδεση μέσω μιας πλακέτας προσαρμογέα που χειρίζεται τη μετατροπή ισχύος και σήματος. • Υποστήριξη έως και 10-δάχτυλα • Τροφοδοσία από τη θύρα GPIO • Καλώδιο ταινίας που συνδέεται με τη θύρα DSI

 <p>Εικόνα 16: Raspberry Pi Camera</p>	<ul style="list-style-type: none"> • Αισθητήρας Sony IMX219 8 megapixel • Υποστηρίζει βίντεο και λήψη 1080p30, 720p60 και VGA90 • Σύνδεση με καλώδιο στη θύρα CSI στο Raspberry Pi • Συμβατά μοντέλα Raspberry 1, 2, 3, 4
---	---

Πίνακας 1: Χαρακτηριστικά μικροεπεξεργαστή και των περιφερειακών του

2.2.2 Υποβιβαστής τάσης HW-319-v6.0

Ο HW-319 [22] είναι μια πλακέτα υπεύθυνη για τον υποβιβασμό και σταθεροποίηση της τάσης. Η συγκεκριμένη πλακέτα υποβιβασμού τάσης χρησιμοποιεί τον ρυθμιστή LM2596, ο οποίος είναι ένα μονολιθικό ολοκληρωμένο κύκλωμα. Έχει την δυνατότητα να μεταφέρει μέχρι και φορτίο 3A. Η σειρά LM2596 λειτουργεί σε συχνότητα μεταγωγής 150 kHz, επιτρέποντας έτσι μικρότερο μέγεθος φίλτρου.

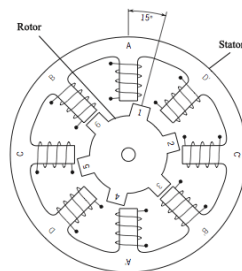
Συσκευή	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 17: Ρυθμιστής Τάσης HW-319-v6</p>	<ul style="list-style-type: none"> • Τάση εισόδου: 4.0V - 40V • Τάση εξόδου: 1.25V - 37V • Ισχύς εξόδου: 20W • Ρεύμα εξόδου: 3A • Θερμοκρασία λειτουργίας -45 °C - +85 °C

Πίνακας 2: Χαρακτηριστικά HW-319-v6

2.2.3 Βηματικοί κινητήρες και οδηγός βηματικών κινητήρων


Οι βηματικοί κινητήρες (stepper motor) [23] κινούνται σε διακριτά βήματα και ανήκουν στην κατηγορία των DC κινητήρων. Σε αντίθεση με τους κινητήρες DC, η κατανάλωση ρεύματος του βηματικού κινητήρα είναι ανεξάρτητη από το φορτίο και έτσι έχουν την τάση να ζεσταίνονται. Αποτελούνται από φάσεις, δηλαδή από πολλά πηνία που οργανώνονται σε ομάδες.


Ο κινητήρας [24] προκειμένου να κινηθεί, θα πρέπει να ενεργοποιηθεί η φάση και έτσι να περιστρέφει, ένα βήμα την φορά (**Εικόνα 18**). Ελέγχονται από οδηγούς που είναι υπεύθυνοι για να την ενεργοποίηση των φάσεων.



Εικόνα 18: Εσωτερική κατανομή βηματικού κινητήρα

Ο οδηγός TB6560 [25] είναι κατασκευασμένος γύρω από το τσιπ Toshiba TB6560AHQ και μπορεί να χρησιμοποιηθεί για την οδήγηση διφασικών διπολικών βηματικών κινητήρων. Με μέγιστο ρεύμα 3A, ο οδηγός TB6560 μπορεί να χρησιμοποιηθεί για τον έλεγχο αρκετά μεγάλων βηματικών κινητήρων. Σαν είσοδο δέχεται τις φάσεις των βηματικών κινητήρων και την τροφοδοσία τους. Δέχεται παλμούς, από τον κεντρικό μικροεπεξεργαστή προκειμένου να ενεργοποιήσει τις φάσεις των κινητήρων και αυτοί να κινηθούν για συγκεκριμένα βήματα ή για συγκεκριμένο χρόνο. Ανάλογα την παλμοσειρά που αποστέλλεται, μπορεί να ρυθμίσει και την ταχύτητα περιστροφής του βηματικού κινητήρα.

Συσκευή	Τεχνικά Χαρακτηριστικά
 <p data-bbox="261 1665 621 1724">Εικόνα 19: Βηματικός κινητήρας 42BYGHW811X1</p>	<ul style="list-style-type: none"> • Μέγεθος σκελετός τύπου: NEMA17 • Βήμα ανά γωνία: 1,8 μοίρας • Τροφοδοσία: 12V • Ρεύμα: 2,5 A / phase • Αντίσταση: 1,25 Ohm / phase • Επαγωγή: 1,8 mH / φάση • Ροπή: 4800g-cm • Αριθμός καλωδίων: 4


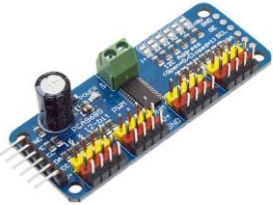
 <p data-bbox="277 464 607 520">Εικόνα 20: Οδηγός βηματικών κινητήρων TB6560</p>	<ul data-bbox="719 226 1463 520" style="list-style-type: none"> • Μέγιστο ρεύμα ανά κανάλι: 3.5A • Τάση λειτουργίας: DC 10V-35V • Ονομαστική έξοδος: $\pm 3,5A$. • Βηματικοί κινητήρες 3A με 2 ή 4 φάσεις / 4 καλώδια ή 6 καλώδια • Λειτουργία διέγερσης: μισό βήμα, 1/8 βήμα, 1/16 βήμα
---	--

Πίνακας 3: Χαρακτηριστικά βηματικού κινητήρα και οδηγού κινητήρων

2.2.4 Σέρβο κινητήρες και οδηγός σέρβο κινητήρων

Οι κινητήρες servo [26] είναι συσκευές που διαθέτουν μεγάλη ακρίβεια κατά την περιστροφή τους. Ο άξονας που διαθέτουν, μπορεί να μετακινηθεί σε διάφορες θέσεις ανάλογα με το κωδικοποιημένο σήμα που λαμβάνουν. Όσο λαμβάνουν το σήμα έχουν την δυνατότητα να κλειδώνουν και να διατηρούν τον άξονα τους σε μια συγκεκριμένη θέση. Οι κινητήρες αυτοί έχουν μικρό μέγεθος αλλά είναι αρκετά ισχυροί. Η ενέργεια που καταναλώνουν είναι ανάλογη με το μηχανικό φορτίο που δέχονται. Για τον έλεγχο τους απαιτείται ένας οδηγός, ο οποίος είναι υπεύθυνος για την αποστολή του κωδικοποιημένου σήματος που λαμβάνουν.

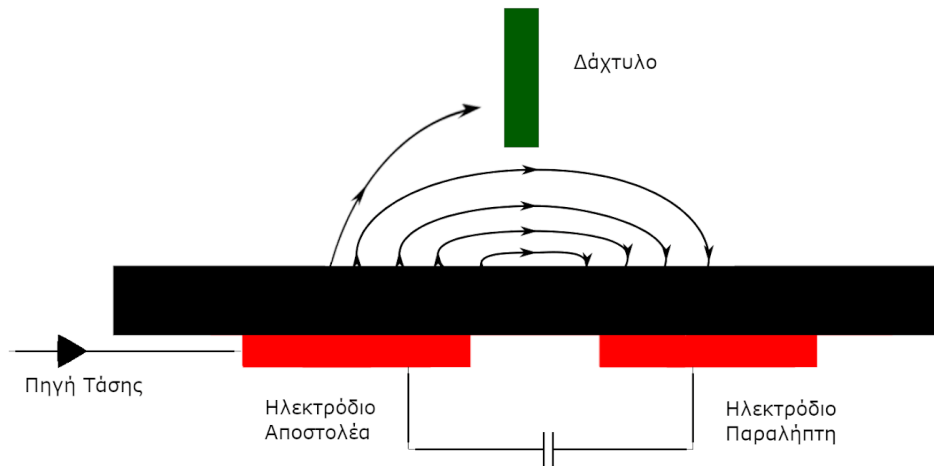
Ο ελεγκτής PCA9685 PWM 16 καναλιών [27] διαθέτει 16 πλήρως προγραμματιζόμενες εξόδους παλμοκωδικής διαμόρφωσης πλάτους (PWM) με ανάλυση 12bit που δίνει συνολικά 4096 προγραμματιζόμενα βήματα. Επιπλέον, η συχνότητα εξόδου και των 16 καναλιών μπορεί να προγραμματιστεί από 24Hz έως 1526Hz. Δίνει επίσης, την δυνατότητα για τον έλεγχο της φωτεινότητας πολλαπλών LED. Το τσιπ TLC5940 τροφοδοτείται με 3,3-5v και η πλακέτα δέχεται τροφοδοσία 5-10v για την τροφοδοσία των 16 καναλιών που διαθέτει. Η μονάδα επικοινωνεί με τον κεντρικό υπολογιστή με το πρωτόκολλο I2C (SDA & SCL) που επιτρέπει την σύνδεση πολλαπλών μονάδων που χρησιμοποιούν το πρωτόκολλο αυτό.

Συσκευή	Τεχνικά Χαρακτηριστικά
 <p data-bbox="321 678 597 705">Εικόνα 21: Servo MG995</p>	<ul data-bbox="751 306 1464 705" style="list-style-type: none"> • Ροπή: 8,5 kgf · cm (4,8 V), 10 kgf · cm (6 V) • Ταχύτητα λειτουργίας: 0,2 s / 60° (4,8 V), 0,16 s / 60° (6 V) • Τάση λειτουργίας: 4,8 V ή 7,2 V • Βηματισμός: 5 μs • Κατασκευή με διπλό ρουλεμάν • Μεταλλικά γράναζια • Εύρος θερμοκρασίας: 0 °C - 55 °C
 <p data-bbox="253 1108 662 1136">Εικόνα 22: Servo controller PCA9685</p>	<ul data-bbox="751 779 1308 1136" style="list-style-type: none"> • Μέγιστο ρεύμα ανά κανάλι: 0.25A • Τάση λειτουργίας: DC 3,3V-5V • Συχνότητα: 40-1000Hz • Αριθμός καναλιών: 16 channel • Ανάλυση: 12 bit • Πρωτόκολλο επικοινωνίας: I2C - PWM • Τάση εισόδου: DC 5V-10V


Πίνακας 4: Χαρακτηριστικά σερβομηχανισμού και του οδηγού PCA9685

2.2.5 Χωρητικός αισθητήρας αφής (Capacitive Touch Sensor) HW-139

Χωρητικότητα [28] ονομάζεται η ικανότητα ενός συστήματος αγωγού να αποθηκεύει ηλεκτρικό φορτίο. Η τιμή της χωρητικότητας είναι συνάρτηση της γεωμετρίας των αγωγών και των διηλεκτρικών υλικών γύρω τους. Η οποιαδήποτε αλλαγή αυτών των στοιχείων αλλάζει και την τιμή της χωρητικότητας. Στην περίπτωση του χωρητικού αισθητήρα αφής όταν ένας δείκτης, δάχτυλο ή γραφίδα, πλησιάσει τις μεταλλικές πλάκες, η αμοιβαία χωρητικότητα μεταξύ των πλακών μειώνεται, καθώς ο δείκτης απορροφά ορισμένες γραμμές ηλεκτρικού πεδίου [29] (Εικόνα 23). Παράλληλα, η αυτοχωρητικότητα και των δύο αγωγών αυξάνεται, καθώς ο δείκτης ενεργεί ως ένας πρόσθετος αγωγός του συστήματος.



Εικόνα 23: Αναπαράσταση παρεμβολής αντικείμενου σε μαγνητικό πεδίο

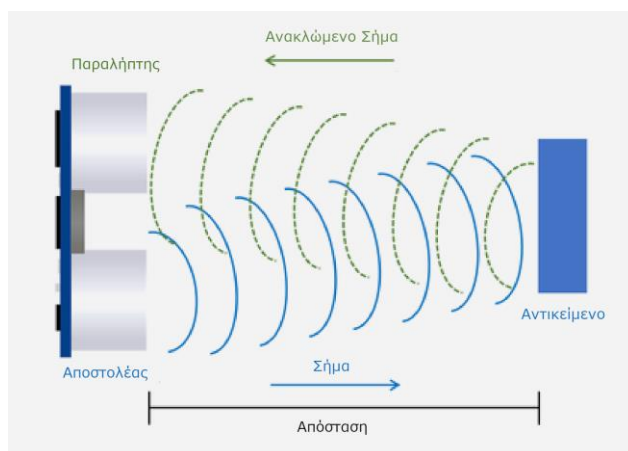
Συσκευή	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 24: Touch sensor HW-139</p>	<ul style="list-style-type: none"> • Τύπος αισθητήρα: Χωρητικό • Τυπική τάση εισόδου: 3.3VDC ή 5VDC • Διεπαφή: Αναλογική • Τάση λειτουργίας: 2.5 - 5.5VDC • Χρόνος απόκρισης: 220mS

Πίνακας 5: Χαρακτηριστικά χωρητικού αισθητήρα αφής HW-139


2.2.6 Αισθητήρας απόστασης υπερήχων HY-SRF05

Ένας αισθητήρας υπερήχων είναι ένα όργανο που μετράει την απόσταση του από ένα αντικείμενο, χρησιμοποιώντας υπερηχητικά κύματα. Πιο συγκεκριμένα, ο αισθητήρας υπερήχων [30] στέλνει ένα ηχητικό κύμα στην συχνότητα των υπερήχων, περίπου στα 40 χιλιάδες Hz. Αυτή η συχνότητα βρίσκεται υψηλότερα στο ηχητικό φάσμα από το εύρος της ανθρώπινης ακοής. Ο μετατροπέας του αισθητήρα λειτουργεί ως δέκτης για την λήψη του υπερηχητικού ήχου. Ο αισθητήρας χρησιμοποιεί μόνο ένα μετατροπέα για την αποστολή του παλμού και την λήψη της

ηχώ του [31] (Εικόνα 25). Η απόσταση προσδιορίζεται από την χρονική διαφορά μεταξύ του διαστήματος αποστολής και λήψης του παλμού του.



Εικόνα 25: Απεικόνιση πρόσκρουσης υπερηχητικού κύματος σε αντικείμενο

Συσκευή	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 26: Ultrasonic HY-SRF05</p>	<ul style="list-style-type: none"> • Εύρος ανίχνευσης: 3cm-4m • Τάση λειτουργίας: 5V (DC) • Στατικό ρεύμα: Λιγότερο από 2mA. • Σήμα εξόδου: Ηλεκτρικό σήμα συχνότητας, υψηλό επίπεδο 5V, χαμηλό επίπεδο 0V. • Γωνία αισθητήρα: Όχι περισσότερο από 15 μοίρες. • Υψηλή ακρίβεια: Έως 0,3 εκατοστά • Επιτρεπτές θερμοκρασίες: -20C - + 60C

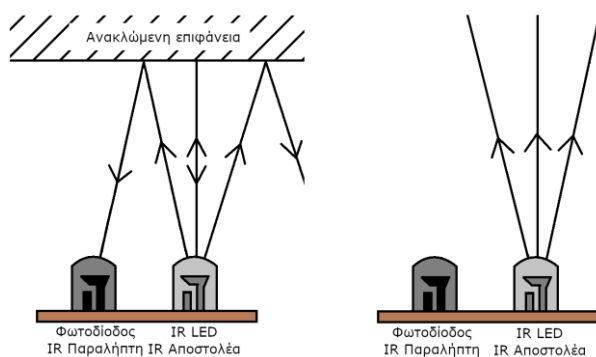
Πίνακας 6: Χαρακτηριστικά υπερηχητικού αισθητήρα HY-SRF05

2.2.7 Αισθητήρας Υπέρυθρων HW-006 V1.3


Υπάρχουν δύο είδη υπέρυθρων αισθητήρων [30], οι ενεργητικοί (πομποί) και οι παθητικοί (δέκτες). Ο υπέρυθρος πομπός είναι μια δίοδος εκπομπής φωτός (LED), ο οποίος εκπέμπει στο φάσμα της υπέρυθρης ακτινοβολίας. Η ζώνη φάσματος της υπέρυθρης ακτινοβολίας βρίσκεται

στην περιοχή των 300 GHz με 400 THz, η ορατή ακτινοβολία στην οποία βρίσκονται όλα τα χρώματα που μπορεί να διακρίνει το ανθρώπινο μάτι τοποθετείται στις συχνότητες 400 - 800 THz, δηλαδή υψηλότερα από την περιοχή υπέρυθρης ακτινοβολίας με αποτέλεσμα να μην είναι ορατό στο ανθρώπινο μάτι. Η διαμόρφωση που χρησιμοποιείται συχνότερα, είναι η διαμόρφωση ΟΟΚ (ON - OFF - KEYING). Ο υπέρυθρος δέκτης ονομάζεται και υπέρυθρος αισθητήρας καθώς ανιχνεύει την ακτινοβολία από έναν υπέρυθρο πομπό. Στην περίπτωση χρήσης συνδυασμού πομπού δέκτη υπέρυθρων, το μήκος κύματος του δέκτη θα πρέπει να ταιριάζει με αυτό του πομπού.

Στην εφαρμογή εντοπισμού εμποδίου με την χρήση των υπέρυθρων αισθητήρων, θέτουμε σε λειτουργία συνεχής εκπομπής υπέρυθρου φωτός τον πομπό και σε λειτουργία λήψης υπέρυθρου φωτός τον δέκτη, με την βοήθεια ενός τελεστικού ενισχυτή (IC Operational Amplifier), ο οποίος χρησιμοποιείται ως συγκριτής τάσης. Όταν το φως που εκπέμπεται από τον υπέρυθρο πομπό προσπίπτει σε εμπόδιο και από εκεί ανακλάται στην φωτοδίοδο του δέκτη, ανάλογα την απόσταση την υφή του υλικού ακόμα και το χρώμα του η υπέρυθρη ακτινοβολία απορροφάται και ανακλάται διαφορετικά, με αποτέλεσμα να υπάρχει διαφορετική τιμή στην έξοδο του δέκτη. Στην περίπτωση που δεν υπάρχει κανένα εμπόδιο μπροστά από τον αισθητήρα, γίνεται εύκολα αντιληπτό ότι δεν θα υπάρχει τιμή στην έξοδο του δέκτη [32] (Εικόνα 27).



Εικόνα 27: Απεικόνιση πρόσκρουσης υπέρυθρου φωτός σε επιφάνεια και στο κενό


Συσκευή	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 28: Infrared HW-006</p>	<ul style="list-style-type: none"> • Αισθητήρας αντανάκλασης υπέρυθρων TCRT5000. • Ανίχνευση της απόστασης ανάκλασης: 1 mm - 25 mm • Τάση λειτουργίας: 5V • Μορφή εξόδου: ψηφιακό σήμα (0 και 1)

Πίνακας 7: Χαρακτηριστικά υπέρυθρου αισθητήρα HW-006

2.2.8 Δίοδοι εκπομπής φωτός (LED) σε σχήμα στεφάνης 12 στοιχείων WS2812B

Το WS2812 [33] είναι μια έξυπνη πηγή φωτισμού στην οποία ενσωματώνεται το κύκλωμα ελέγχου. Περιλαμβάνει σύνδεση δεδομένων ψηφιακής θύρας. Το πρωτόκολλο μεταφοράς δεδομένων χρησιμοποιεί έναν τρόπο επικοινωνίας NRZ (Not Return Zero). Μετά την ενεργοποίηση κάποιου pixel, η θύρα λαμβάνει δεδομένα από τον ελεγκτή. Το πρώτο εικονοστοιχείο, συλλέγει αρχικά δεδομένα 24bit και μετά τα αποστέλλει στο επόμενο εικονοστοιχείο. Το τσιπ ελέγχου είναι ενσωματωμένο σε κάθε LED, έτσι γίνεται πιο απλό το απαιτούμενο κύκλωμα ελέγχου και του προδίδει μικρό όγκο.

Η στεφάνη στοιχείων (LED ring) αποτελείται από 12 εξαιρετικά φωτεινά LED με ενσωματωμένο το τσιπ WS2812. Είναι διατεταγμένα σε κύκλο με εξωτερική διάμετρο 37mm. Η κατανάλωση ενέργειας του κάθε LED ανέρχεται στα 18mA και τροφοδοτείται με 5V συνεχούς ρεύματος.

Συσκευή	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 29: LED ring WS2812B</p>	<ul style="list-style-type: none"> • Πηγή φωτός: 5050 RGB • Μοντέλο IC: WS2812B • Τάση εισόδου 5V • Din: Εισαγωγή σήματος δεδομένων • Dout: Έξοδος σήματος δεδομένων

Πίνακας 8: Χαρακτηριστικά δίοδων εκπομπής φωτός σε στεφάνη WS2812B

2.3 Σύνοψη

Σε αυτό το κεφάλαιο αναφέρθηκε το θεωρητικό υπόβαθρο των εργαλείων που βοήθησαν στην δημιουργία του λογισμικού καθώς και χαρακτηριστικά των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη του ρομποτικού συστήματος. Στο κεφάλαιο που ακολουθεί θα αναλυθούν οι απαιτήσεις, οι περιπτώσεις χρήσης και οι ενεργειακές ανάγκες του συστήματος. Τέλος, γίνεται παρουσίαση για την μελέτη του σώματος του ρομπότ.

Κεφάλαιο 3 – Δομικά μέρη συστήματος

Στο κεφάλαιο αυτό, θα παρατεθούν οι απαιτήσεις του συστήματος, θα αναλυθούν οι περιπτώσεις χρήσης, τα υλικά που χρησιμοποιήθηκαν, η συνδεσμολογία και οι ενεργειακές προϋποθέσεις του συστήματος. Τέλος, παρουσιάζεται η μελέτη για την κατασκευή του σώματος του ρομπότ.

3.1 Απαιτήσεις

Ένα από το πιο σημαντικά κομμάτια για την επιτυχημένη δημιουργία ενός συστήματος, είναι ο καθορισμός των απαιτήσεων του. Το λογισμικό προορίζεται για χρήση κυρίως από παιδιά και πρέπει να διαθέτει φιλικό περιβάλλον. Πρέπει να είναι αρκετά απλό και να διαθέτει χρώματα που είναι ελκυστικά στα παιδιά. Η ενσωμάτωση της μητρικής γλώσσας αποτελεί βασικό χαρακτηριστικό των συστημάτων που προορίζονται για παιδιά.

Ο σχεδιασμός του ρομπότ έχει μελετηθεί προκειμένου να είναι ασφαλής και θελκτικός προς στα παιδιά. Ειδικότερα, το ρομπότ έχει λείες όλες τις ακμές του, προκειμένου να μην είναι κοφτερές και να μην υπάρχει κίνδυνος τραυματισμού. Το παιδί θα πρέπει να νιώθει οικεία με το ρομπότ, να μην τρομάζει με την εμφάνιση ή τις κινήσεις του. Ο σχεδιασμός του παραπέμπει σε εικόνες που έχει το κάθε παιδί για την μορφή ενός ρομποτικού συστήματος (ανθρωποειδές).

Η υποδομή για την διασύνδεση του ρομπότ με το πληροφοριακό σύστημα γίνεται με όλα τα πρότυπα ασφάλειας, προκειμένου να αποφευχθούν κακόβουλες ενέργειες. Πιο συγκεκριμένα, η διασύνδεση υλοποιείται με το πρωτόκολλο μεταφοράς υπερκειμένου (Hypertext Transfer Protocol - HTTP). Προκειμένου να επιτευχθεί με επιτυχία η διασύνδεση με το πληροφοριακό σύστημα είναι απαραίτητη η σύνδεση του ρομπότ στο διαδίκτυο και να ακολουθούνται από πλευράς του πληροφοριακού συστήματος όλα τα πρότυπα ασφαλείας.

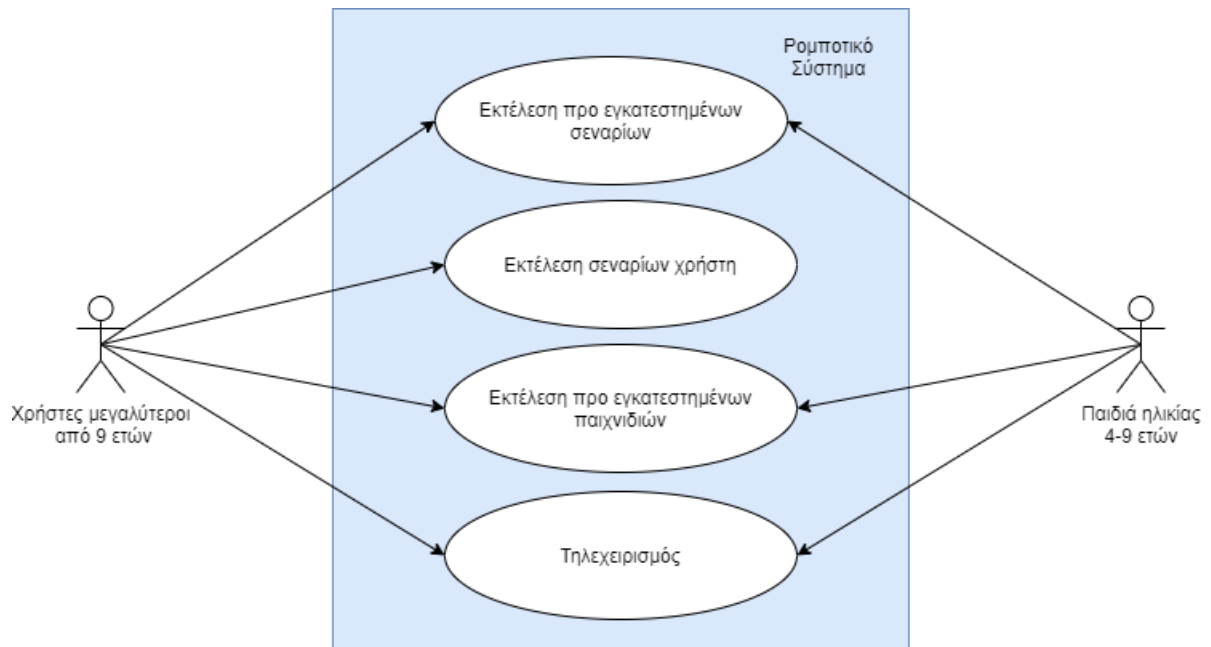
Για την πλήρη εκμετάλλευση του ρομπότ, δίνεται η δυνατότητα, σε χρήστες μεγαλύτερης ηλικίας με απλές γνώσεις προγραμματισμού, να αναπτύξουν και να εκτελέσουν τα δικά τους σενάρια. Το σύστημα διαθέτει ελέγχους ασφαλείας, προκειμένου να μην επιτρέπεται στον χρήστη να βλάψει το ρομπότ.

Το σύστημα απαιτείται να είναι επεκτάσιμο και συντηρήσιμο, προκειμένου να επιτευχθούν τα παραπάνω θα πρέπει το ρομπότ τόσο σε επίπεδο υλικού όσο και σε επίπεδο λογισμικού να είναι ικανό να δεχτεί και να υποστηρίξει πληθώρα σεναρίων.

3.2 Περιπτώσεις χρήσης

Στην συγκεκριμένη ενότητα παρουσιάζονται οι περιπτώσεις χρήσης (**Εικόνα 30**) του συστήματος. Το σύστημα υποστηρίζει δύο ειδών χρήστες, παιδιά ηλικίας τεσσάρων με εννέα ετών και άτομα μεγαλύτερης ηλικίας, που έχουν την δυνατότητα ανάπτυξης των δικών τους σεναρίων.

- **Η εκτέλεση των προ εγκατεστημένων σεναρίων** δίνει την δυνατότητα να αλληλεπιδράσουν τα παιδιά με το ρομπότ.
- **Στους χρήστες μεγαλύτερης ηλικίας, δίνεται η επιλογή συγγραφής και εκτέλεσης σεναρίων στο ρομπότ.** Με την χρήση πληροφοριακού συστήματος με το οποίο έχει την δυνατότητα διασύνδεσης γίνεται η φόρτωση των σεναρίων αυτών. Στην περίπτωση που ένας μεγαλύτερης ηλικίας χρήστης δεν είναι αρκετά εξοικειωμένος και έμπειρος με τον προγραμματισμό αντίστοιχων συστημάτων, δίνεται η δυνατότητα προγραμματισμού με απλές συναρτήσεις, όπου με την εκτέλεση τους είναι εφικτή η υλοποίηση των δικών τους σεναρίων.
- **Τα προ εγκατεστημένα εκπαιδευτικά παιχνίδια** που διαθέτει δίνουν την δυνατότητα στο παιδί να ασχοληθεί και να κατανοήσει καλύτερα τα μαθήματα, όπως είναι η γλώσσα, η φυσική και τα μαθηματικά. Τα παιχνίδια διαθέτουν διάφορα επίπεδα δυσκολίας και χωρίζονται σε δύο βασικές ηλικιακές κατηγορίες.
- **Δυνατότητα τηλεχειρισμού του ρομπότ.** Μέσω της υποστήριξης που διαθέτει το σύστημα για διασύνδεση με ειδική εφαρμογή, ο χρήστης θα έχει πρόσβαση στην κάμερα, καθώς και στους διάφορους κινητήρες του ρομπότ.



Εικόνα 30: Διάγραμμα περιπτώσεων χρήσης

3.3 Υλικό μέρος

Στην παρούσα ενότητα, αναλύονται τα υλικά που χρησιμοποιήθηκαν για την κατασκευή του ρομπότ. Βασικό κομμάτι του ρομπότ είναι ο μικροεπεξεργαστής που επιλέχθηκε και πρόκειται για ένα SBC (Single Board Computer) [34] το Raspberry Pi 4. Στην πλακέτα, συνδέονται η οθόνη αφής 7 ιντσών, όλα τα αισθητήρια όργανα του ρομπότ καθώς και οι κινητήρες που είναι υπεύθυνοι για την κίνηση των χεριών, του κεφαλιού και των ροδών.

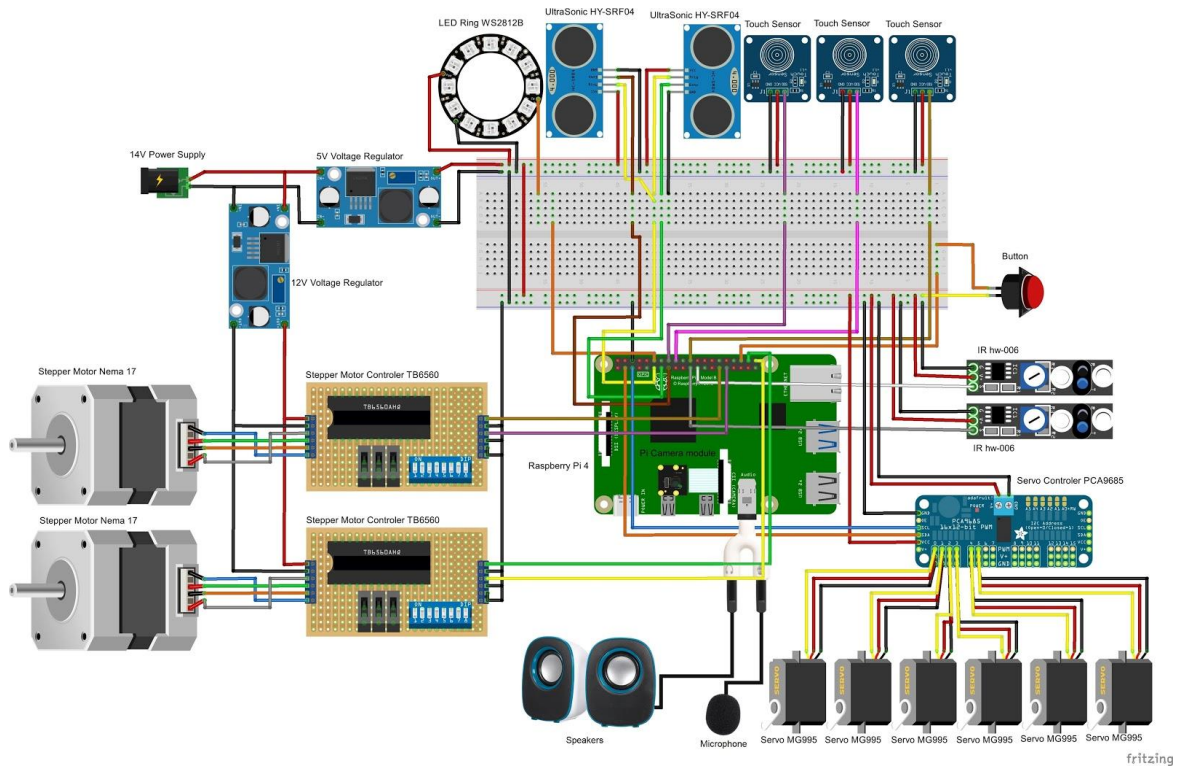
Συγκεκριμένα, το ρομπότ αποτελείται:

- 1x Raspberry Pi 4 4GB ram & 32GB SD card
- 1x Raspberry Pi 3 touchscreen 7 inch
- Raspberry Pi camera
- 6x Metal gear servo
- 1x 16 Channel PWM servo controller
- 2x Nema 17 stepper motor
- 2x TB6560 Stepper motor controller
- 3x Capacitive Touch sensor
- 2x Ultrasonic distance sensor
- 2x Infrared sensor
- 2x Speakers 2.8W
- 1x Led ring ws2812b 12 led
- 1x Button with 12V led
- 1x Microphone
- 2x Voltage regulators

- 1x Power on/off switch
- 1x Lipo battery 145CA
- Βίδες 3mm με παξιμάδια
- Ντίζες 4mm με παξιμάδι

3.4 Συνδεσμολογία

Στην ενότητα αυτή παρουσιάζεται τόσο η συνδεσμολογία (Εικόνα 31) όσο και η τροφοδοσία των κινητήρων και των αισθητήριων οργάνων του ρομπότ.



Εικόνα 31: Κυκλωματική απεικόνιση του συστήματος

Το ρομπότ τροφοδοτείται με μια πηγή τάσης 14V τα οποία οδηγούνται σε δύο υποβιβαστές τάσης (Step Down Voltage Regulator). Από τον πρώτο εξάγεται μια τάση στα 5V, η οποία τροφοδοτεί όλα τα ηλεκτρονικά στοιχεία, τα σέρβο και τον μικροελεγκτή. Ο δεύτερος υποβιβαστής τάσης εξάγει τάση 12V, προκειμένου να τροφοδοτηθούν οι βηματικοί κινητήρες. Οι δύο βηματικοί κινητήρες που διαθέτει το ρομπότ για την περιήγηση του στον χώρο ελέγχονται από

έναν ελεγκτή TB6560 ο καθένας. Στην αριστερή πλευρά του οδηγού TB6560 συνδέονται η εξωτερική τάση 12V (V+) για την λειτουργία των βηματικών κινητήρων, καθώς και η γείωση του (GND). Στην συνέχεια συνδέονται η θετική πρώτη φάση (A+), η αρνητική πρώτη φάση (A-), η θετική δεύτερη φάση (B+) του βηματικού κινητήρα και τέλος η αρνητική δεύτερη φάση (B-) του. Στην απέναντι πλευρά της πλακέτας του βηματικού οδηγού υπάρχουν ζεύγη θετικών και αρνητικών σημάτων για την επικοινωνία των κινητήρων και του μικροεπεξεργαστή. Όλα τα αρνητικά μέρη των ζευγαριών (CLK-, CH-, EN-) οδηγούνται στην γείωση, επίσης τον πέμπτο ακροδέκτη (EN+) αποτελεί το σήμα ενεργοποίησης των κινητήρων. Ο πρώτος ακροδέκτης (CLK+) αποτελεί την εντολή επιθυμητών βημάτων του κινητήρα και ο τρίτος ακροδέκτης (CH+) δηλώνει την διεύθυνση περιστροφής του βηματικού κινητήρα. Όσον αφορά τους διακόπτες (Switch) που διαθέτει η πλακέτα του οδηγού βηματικού κινητήρα, ρυθμίζουν το μέγιστο ρεύμα στην είσοδο των κινητήρων καθώς και τις ρυθμίσεις που αφορούν την ευκρίνεια βήματος.

Το ρομπότ στα πλαίσια της ασφαλούς και σωστής περιήγησης του στον χώρο διαθέτει τέσσερις αισθητήρες. Δύο εξ' αυτών ευθύνονται για τις μπροστινές κινήσεις και άλλοι δύο για τις κινήσεις που εκτελεί το ρομπότ προς τα πίσω. Στην κάτω μεριά του ρομπότ στραμμένοι προς το έδαφος, βρίσκονται οι δύο αισθητήρες υπερέυθρων (InfraRed), ένας στην μπροστινή πλευρά και ένας στην πίσω. Με αυτούς, το ρομπότ διακρίνει πότε υπάρχει έδαφος για να κινηθεί και πότε όχι. Η σύνδεση τους με τον μικροελεγκτή πραγματοποιείται με έναν ακροδέκτη έκαστος και τροφοδοτούνται από την έξοδο 5V του υποβιβαστή τάσης (Step Down Voltage Regulator). Επίσης, το ρομπότ διαθέτει ακόμη δύο υπερηχητικούς αισθητήρες (UltraSonic), οι οποίοι βρίσκονται ένας στην μπροστινή πλευρά και ένας στην πίσω για τον εντοπισμό εμποδίων. Οι συγκεκριμένοι αισθητήρες τροφοδοτούνται με 5V από τον υποβιβαστή τάσης (Step Down Voltage Regulator) και συνδέονται με τον μικροεπεξεργαστή με την χρήση δυο ακροδεκτών.

Στα πλαίσια της αλληλεπίδρασης του ρομπότ με τους χρήστες προστέθηκαν τρεις αισθητήρες επαγωγικής επαφής (Capacitive Touch Sensor), ένα κουμπί και ορισμένοι δίοδοι εκπομπής φωτός (LED) σε σχήμα στεφάνης 12 στοιχείων. Οι τρεις αισθητήρες χωρητικής επαγωγής (Capacitive Touch Sensor) απαιτούν 5V τροφοδοσίας και ένα ακροδέκτη για την σύνδεση τους με τον μικροεπεξεργαστή. Το κουμπί είναι ένας διακόπτης με δύο ακροδέκτες, ο πρώτος συνδέεται στην τροφοδοσία 5V και ο δεύτερος σε έναν ακροδέκτη γενικού σκοπού του

Raspberry Pi 4. Τέλος, η στεφάνη διόδων εκπομπής φωτός (LED) τροφοδοτείται με τάση εισόδου 5V και δέχεται εντολές μέσω ενός ακροδέκτη από τον μικροεπεξεργαστή.

Ο χειρισμός των σέρβο κινητήρων από τον μικροεπεξεργαστή διεκπεραιώνεται σύμφωνα με το πρωτόκολλο επικοινωνίας Inter-Integrated Circuit (I2C). Το συγκεκριμένο πρωτόκολλο χρησιμοποιεί δύο ακροδέκτες, ένα για τα δεδομένα (SDA) και ένα για τον συγχρονισμό του ρολογιού (SDL). Ο ελεγκτής PCA9685 PWM, ο οποίος είναι υπεύθυνος για τον χειρισμό των σέρβο κινητήρων, τροφοδοτείται από τον υποβιβαστή τάσης (Step Down Voltage Regulator) με 5V. Τέλος, οι σέρβο κινητήρες συνδέονται στην κάτω μεριά της πλακέτας σε κάθετη διάταξη μέσω τριών ακροδεκτών PWM, V++ και GND. Στο ρομπότ χρησιμοποιήθηκαν έξι σέρβο κινητήρες MG995 γωνίας 0-180 μοιρών.

Για την φωνητική και ακουστική επικοινωνία του χρήστη με το ρομπότ, χρησιμοποιήθηκαν ένα μικρόφωνο βύσματος (jack) 3.5 mm, και ένα σετ ηχείων με είσοδο βύσμα (jack) 3.5 mm. Προκειμένου τα δύο αυτά εξαρτήματα να συνδεθούν στην θύρα εισόδου/εξόδου ήχου (audio jack) της πλακέτας, χρησιμοποιήθηκε ένας σύνδεσμος (audio splitter).

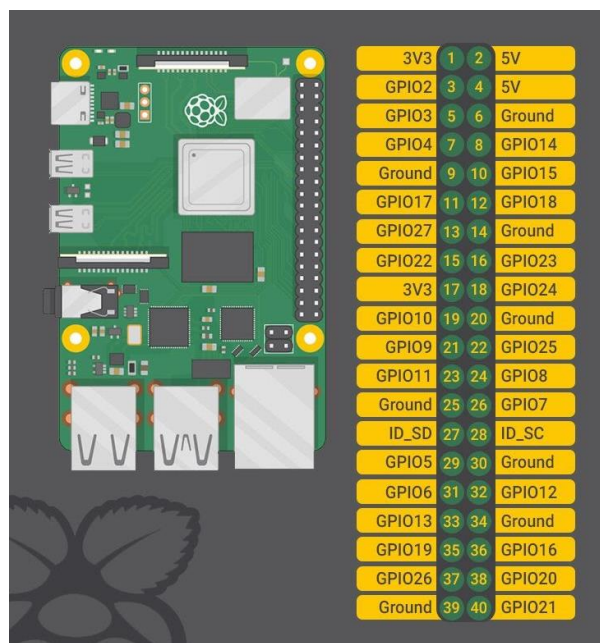
Η οθόνη αφής 7 ιντσών συνδέθηκε στον μικροεπεξεργαστή μέσω της θύρας DSI (Display Serial Interface) που διαθέτει το Raspberry Pi 4. Η κάμερα Pi Camera συνδέθηκε στον μικροεπεξεργαστή στην θύρα CSI (Camera Serial Interface), η οποία βρίσκεται και αυτή πάνω στο Raspberry Pi 4.

Όλοι οι ακροδέκτες (**Εικόνα 32**) που συνδέουν τα επιμέρους στοιχεία συνοψίζονται στον παρακάτω πίνακα (**Πίνακας 9**).

A/A	Εξάρτημα	Τροφοδοσία (V)	Pin του Raspberry Pi 4
1	Οδηγός βηματικού κινητήρα TB6560 No1	12	29, 31
2	Οδηγός βηματικού κινητήρα TB6560 No2	12	38, 40
3	Υπέρυθρος αισθητήρας hw-006 V1.3 No1	5	21
4	Υπέρυθρος αισθητήρας hw-006 V1.3 No2	5	23
5	Υπέρηχος αισθητήρας hy-srf05 No1	5	11, 13

6	Υπέρηχος αισθητήρας hy-srf05 No2	5	11, 15
7	Χωρητικός αισθητήρας αφής hw-139 No1	5	16
8	Χωρητικός αισθητήρας αφής hw-139 No2	5	18
9	Χωρητικός αισθητήρας αφής hw-139 No3	5	22
10	Κουμπί	5	36
11	Στεφάνι 12 διόδων εκπομπής φωτός ws2812b	5	12
12	Οδηγός σέρβο κινητήρων PCA9685	5	3, 5

Πίνακας 9: Πίνακας εισόδων/ εξόδων μικροεπεξεργαστή



Εικόνα 32: Είσοδοι/ έξοδοι μικροεπεξεργαστή

3.5 Ενεργειακές απαιτήσεις

Για την σωστή λειτουργία του ρομπότ έγινε μελέτη για την απαραίτητη ενέργεια του συστήματος, προκειμένου τόσο ο μικροεπεξεργαστής Raspberry Pi 4 όσο οι κινητήρες και αισθητήρες που διαθέτει να είναι σε θέση να λειτουργούν απροβλημάτιστα. Για την τροφοδοσία

του συστήματος επιλέχθηκε μια μπαταριά λιθίου 14V και 8Ah. Απαίτηση του ρομποτικού συστήματος ήταν το μικρό βάρος, η μεγάλη αυτονομία αλλά και ο μικρός χρόνος επαναφόρτισης.

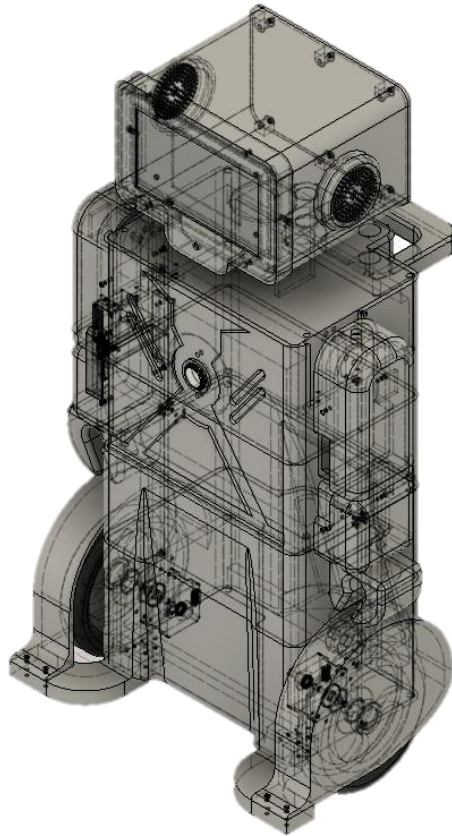
Οι ενεργειακές απαιτήσεις του μικροεπεξεργαστή που χρησιμοποιήθηκε ανέρχονται στα 3A. Οι βηματικοί κινητήρες που είναι υπεύθυνοι για την κίνηση του ρομπότ στον χώρο τροφοδοτούνται με 12V και καταναλώνουν 2,5A έκαστος σε μέγιστο φορτίο. Οι σέρβο κινητήρες που είναι υπεύθυνοι για την κίνηση των χεριών αλλά και του κεφαλιού τροφοδοτούνται με 5V και καταναλώνουν σε μέγιστο φορτίο 1,2A. Οι αισθητήρες και οι ενδεικτικές λυχνίες του ρομπότ καταναλώνουν κατά μέσο όρο 400mA.

3.6 Σχεδιασμός και εμφάνιση ρομπότ

Για τον σχεδιασμό της εμφάνισης του ρομπότ (**Εικόνα 33**) μελετηθήκαν οι λειτουργικές ανάγκες, η ασφάλεια, αλλά και η μελλοντική ανάπτυξη του. Στην συνέχεια σχεδιάστηκαν με την βοήθεια συνεργάτη του εργαστήριου Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων, προκειμένου να σχεδιαστούν τα πλαστικά του ρομπότ.

- Το σασί του ρομπότ σχεδιάστηκε με τρόπο ώστε να εξασφαλίσει την σταθερότητα, την ανθεκτικότητα αλλά και να επιτευχθεί το απαραίτητο για την κίνηση, κέντρο βάρους.
- Το εσωτερικό της κατασκευή προστατεύεται από πλαστικά καλύμματα για να αποτραπεί η πρόσβαση στο εσωτερικό του, στο οποίο φιλοξενούνται τα ηλεκτρονικά του μέρη.
- Το ρομποτικό σύστημα διαθέτει δύο βραχίονες με διπλές αρθρώσεις. Έχουν σχεδιαστεί με τέτοιο τρόπο ώστε στο ρομπότ να δίνεται η δυνατότητα προσθήκης διαφόρων εξαρτημάτων στα χέρια του, με τα οποία διευρύνονται οι λειτουργίες και τα σενάρια τα οποία θα μπορεί να εκτελέσει.
- Το κεφάλι έχει την δυνατότητα να κινηθεί σε δύο άξονες, τον οριζόντιο άξονα (δεξιά - αριστερά) και τον κάθετο άξονα (πάνω - κάτω). Με αυτόν τον τρόπο μπορεί να εκφράζει καλύτερα τα συναισθήματα (συμφωνία, άρνηση, εκνευρισμός κλπ.) που πηγάζουν από τα διάφορα σενάρια που εκτελεί.
- Η προσθήκη χειρολαβής κρίθηκε αναγκαία για την απλοποίηση της διαδικασίας μεταφοράς του.

Όλα τα κομμάτια του ρομπότ δημιουργήθηκαν με την διαδικασία της ταχείας πρωτοτυποποίησης. Το πολυγαλακτικό οξύ ή κοινός PLA αποτελεί το υλικό που χρησιμοποιήθηκε για την κατασκευή του πρωτότυπου μοντέλου. Πρόκειται για ένα βιοδιασπώμενο και βιοδραστικό θερμοπλαστικό.



Εικόνα 33: Τρισδιάστατο σχέδιο εμφάνισης ρομπότ

3.7 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκαν οι απαιτήσεις που πρέπει να έχει ένα ολοκληρωμένο ρομποτικό σύστημα. Έγινε αναφορά στις δύο κατηγορίες χρηστών και στις δυνατότητες που τους παρέχονται. Περιγράφηκε η μέθοδος συνδεσμολογίας των περιφερειακών με τον μικροεπεξεργαστή, οι ενεργειακές απαιτήσεις και οι απαιτήσεις που τέθηκαν για τον σχεδιασμό του σώματος του ρομπότ.

Κεφάλαιο 4 – Ανάλυση Λογισμικού

Αφού περιεγραφήκαν οι απαιτήσεις του απαραίτητου υλικού μέρους του συστήματος, στο παρόν κεφάλαιο επεξηγείται η ολοκλήρωση του λογισμικού. Αναλύεται το λογισμικό που υλοποιήθηκε και εγκαταστάθηκε στον μικροεπεξεργαστή μας. Τα προ εγκατεστημένα σενάρια και ιδιότητες που προσδίδουν στο ρομπότ. Στην συνέχεια, γίνεται αναφορά στην δυνατότητα διασύνδεσης του ρομπότ με το πληροφοριακό σύστημα και πως θα πρέπει να επιτευχθεί η μεταξύ τους επικοινωνία.

4.1 Λογισμικό μικροεπεξεργαστή

Το λειτουργικό σύστημα που επιλέχθηκε για τον μικροεπεξεργαστή μας είναι το Raspberry Pi OS. Πρόκειται για ένα λειτουργικό σύστημα με βάση το λειτουργικό Debian για το Raspberry Pi. Η εγκατάσταση της διανομής Raspbian Buster έγινε σε μια MicroSD 3 GB.

Για την σωστή λειτουργία του λογισμικού που αναπτύχθηκε, χρειάστηκε η παραμετροποίηση του μικροεπεξεργαστή προκειμένου να είναι σε θέση να αξιοποιηθούν τόσο εισοδοί όσο και οι εξοδοί που διαθέτει η πλακέτα. Στην συνέχεια, έγινε η εγκατάσταση των απαραίτητων βιβλιοθηκών και εργαλείων. Οι βιβλιοθήκες που απαιτήθηκαν για τον εντοπισμό προσώπων και αντικειμένων είναι το *OpenCV python library* και η *Caffe model deep learning framework*. Η εγκατάσταση της βιβλιοθήκης *PyQt v5* και του σχεδιαστή *QtDesigner* κρίθηκαν αναγκαία για την εκτέλεση του γραφικού περιβάλλοντος. Υπεύθυνες για την διαχείριση των ήχων του συστήματος είναι οι βιβλιοθήκες *Pulse audio* και *Simple audio*. Ο χειρισμό των σέρβο κινητήρων και της στεφάνης LED έγινε με την βοήθεια των *Adafruit-circuitPython-servokit* και *Adafruit-circuitPython-neopixel* αντίστοιχα. Η εκμετάλλευση των ακροδεκτών εισόδου και εξόδου (GPIO) έγινε με την προ εγκατεστημένη βιβλιοθήκη *gpiozero*. Η σύνδεση του ρομπότ με το διαδίκτυο πραγματοποιήθηκε με το λογισμικό *ComitUp* που προσφέρει δυνατότητα σύνδεσης στο διαδίκτυο μέσω άλλης συσκευής. Τέλος, το λογισμικό *GCompris* εγκαταστάθηκε με σκοπό την προσθήκη πληθώρας εκπαιδευτικών παιχνιδιών στο ρομποτικό σύστημα.

4.2 Συναρτήσεις

Στην παρούσα ενότητα θα παρουσιαστούν και θα αναλυθούν οι κώδικες που είναι υπεύθυνοι για το κεντρικό κουμπί του ρομπότ, την κίνηση των βηματικών κινητήρων, σέρβο κινητήρων, την διαχείριση των αισθητήριων οργάνων και της στεφάνης led.

4.2.1 Κουμπί πολλαπλών χρήσεων

Το ρομπότ διαθέτει στο στήθος του ένα στρογγυλό, μεγάλο σε μέγεθος, κουμπί πολλαπλών χρήσεων. Βασική λειτουργία του είναι να βοηθάει στην πλοήγηση του μενού της εφαρμογής. Με παρατεταμένο πάτημα δύο δευτερολέπτων επιστρέφει στην αρχική οθόνη και παράλληλα σταματάει όλες τις λειτουργίες του ρομπότ θέτοντας σε αρχική κατάσταση τους κινητήρες, το κεφάλι και τα χέρια. Τέλος, με παρατεταμένο πάτημα οκτώ δευτερολέπτων κλείνει το ρομπότ.

4.2.2 Αισθητήρες απόστασης

Το ρομπότ στο σώμα του, διαθέτει δύο υπερηχητικούς αισθητήρες που είναι υπεύθυνοι για τον εντοπισμό αντικειμένων, καθώς και δύο υπέρυθρους αισθητήρες που εντοπίζουν αν το ρομπότ βρίσκεται πάνω σε κάποια επιφάνεια. Οι αισθητήρες αυτοί λειτουργούν παράλληλα με τους κινητήρες προκειμένου να αποφευχθεί σύγκρουση του ρομπότ με κάποιο εμπόδιο ή κάποια πιθανή πτώση του ρομπότ.

Η συνάρτηση `ultra_sonic_front()` και `ultra_sonic_back()` (**Εικόνα 34**) δέχονται σαν όρισμα την απόσταση, για να σταματήσει το ρομπότ στην περίπτωση που εντοπίσει κάποιο εμπόδιο. Για λόγους ασφάλειας η τιμή της απόστασης δεν μπορεί να είναι μικρότερη από πέντε εκατοστά. Στην περίπτωση που το ρομπότ εντοπίσει κάποιο εμπόδιο επιστρέφει μια ψευδής boolean τιμή προκειμένου να ακινητοποιηθούν οι κινητήρες. Όταν η τιμή αλλάξει οι κινητήρες θα συνεχίσουν προκειμένου να ολοκληρώσουν την κίνηση τους.

<pre>def ultra_sonic_front(dist): try: GPIO.output(TRIG, GPIO.HIGH) time.sleep(0.00001) GPIO.output(TRIG, GPIO.LOW) while GPIO.input(ECHO)==0: pulse_start = time.time() while GPIO.input(ECHO)==1: pulse_end = time.time() pulse_duration = pulse_end - pulse_start distance = pulse_duration * 17150 distance = round(distance + 1.15, 2) print ("distance front:",distance,"cm") if distance>dist: return True elif return False except KeyboardInterrupt: GPIO.cleanup() return False</pre>	<pre>def ultra_sonic_back(dist): try: GPIO.output(TRIG, GPIO.HIGH) time.sleep(0.00001) GPIO.output(TRIG, GPIO.LOW) while GPIO.input(ECHO2)==0: pulse_start2 = time.time() while GPIO.input(ECHO2)==1: pulse_end2 = time.time() pulse_duration2 = pulse_end2 - pulse_start2 distance2 = pulse_duration2 * 17150 distance2 = round(distance2 + 1.15, 2) print ("distance back:",distance2,"cm") if distance2>dist: return True elif return False except KeyboardInterrupt: GPIO.cleanup() return False</pre>
---	--

Εικόνα 34: Συναρτήσεις ελέγχου υπερηχητικών αισθητήρων

Η συνάρτηση `ir_front()` και `ir_back()` (Εικόνα 35) ελέγχουν τους υπέρυθρους αισθητήρες του ρομπότ που έχουν τοποθετηθεί στο κάτω μέρος με σκοπό να εντοπίζουν την αλλαγή επιφάνειας, για παράδειγμα το κενό. Οι συναρτήσεις, στην περίπτωση αλλαγής επιφάνειας, επιστρέφουν μια ψευδής boolean τιμή προκειμένου να ακινητοποιηθούν οι κινητήρες.

<pre>def ir_front(): try: if GPIO.input(SensorPin) == GPIO.LOW: print ("On air") return False else: print ("On table") return True except KeyboardInterrupt: GPIO.cleanup() return False</pre>	<pre>def ir_back(): try: if GPIO.input(SensorPin2) == GPIO.LOW: print ("On air 2") return False else: print ("On table 2") return True except KeyboardInterrupt: GPIO.cleanup() return False</pre>
--	--

Εικόνα 35: Συναρτήσεις ελέγχου υπέρυθρων αισθητήρων

4.2.3 Επαγωγικοί αισθητήρες αφής

Το ρομπότ διαθέτει τρεις επαγωγικούς αισθητήρες αφής. Ένας έχει τοποθετηθεί στο κεφάλι και άλλοι δύο στους ώμους. Η συνάρτηση `touch()` (Εικόνα 36) δέχεται σαν όρισμα το `id` (1-3) του αισθητήρα και επιστρέφει μια αληθής τιμή `boolean` όταν ενεργοποιείται ο αισθητήρας.

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(24, GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(25, GPIO.IN,pull_up_down=GPIO.PUD_UP)

def touch(t_id):
    if (GPIO.input(t_id) == True):
        print ("You touch me ", t_id)
        return True
    else:
        return False
```

Εικόνα 36: Συνάρτηση ελέγχου επαγωγικών αισθητήρων αφής

4.2.4 Δίοδοι εκπομπής φωτός

Για λειτουργία της στεφάνης led χρησιμοποιήθηκε η βιβλιοθήκη `board` και `neopixel` (Εικόνα 37). Η συνάρτηση `rainbow_cycle()` δέχεται σαν όρισμα την ταχύτητα της κυκλική εναλλαγής των χρωμάτων των led. Η συνάρτηση `set_led()` αξιοποιεί την κλάση της βιβλιοθήκης `neopixel pixels.fill` και δέχεται σαν ορίσματα τον κωδικό του χρώματος σε RGB (Red, Green, Blue).

```
pixels = neopixel.NeoPixel(
    pixel_pin, num_pixels, brightness=1, auto_write=False, pixel_order=ORDER
)

def rainbow_cycle(wait):
    for j in range(255):
        for i in range(num_pixels):
            pixel_index = (i * 256 // num_pixels) + j
            pixels[i] = wheel(pixel_index & 255)
        pixels.show()
        time.sleep(wait)

def set_led(R,G,B):
    pixels.fill((R, G, B))
    pixels.show()
```

Εικόνα 37: Συνάρτηση διόδων εκπομπής φωτός

4.2.5 Σέρβο κινητήρες

Για την σωστή διατήρηση των κινητήρων υλοποιήθηκαν δύο συναρτήσεις η `servo_move()` και η συνάρτηση `servo_speed()` (Εικόνα 38). Η πρώτη συνάρτηση δέχεται σαν ορίσματα τον αριθμό (`id`) του σέρβο κινητήρα που θέλουμε να ενεργοποιήσουμε και την θέση του σε μοίρες (`ang`). Η δεύτερη συνάρτηση δέχεται παραμέτρους το σέρβο που θέλουμε να κινήσουμε (`id`), την θέση του σε μοίρες (`ang`), το βήμα αύξησης των μοιρών (`step`) και τον χρόνο αύξησης των μοιρών (`slp`) σε δευτερόλεπτα. Οι αποδεκτές γωνίες των σέρβο κινητήρων είναι από μηδέν μέχρι εκατόν ογδόντα.

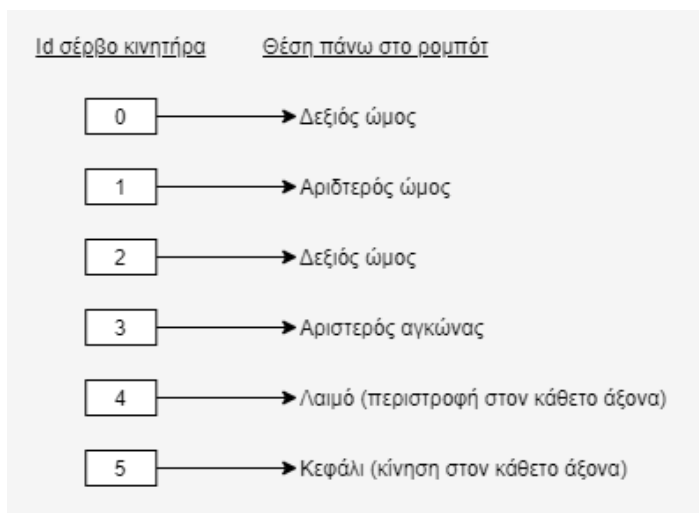
```
from time import sleep
from adafruit_servokit import ServoKit

kit = ServoKit(channels=8)

def servo_move(id, ang):
    kit.servo[id].angle = ang
    return True

def servo_speed(id, ang, step, slp):
    for angle in range(id, ang, step): # 0 - 180 degrees, 5 degrees at a time.
        my_servo.angle = angle
        time.sleep(slp)
```

Εικόνα 38: Συνάρτηση ελέγχου σέρβο κινητήρων



Εικόνα 39: Αντιστοιχία σέρβο κινητήρων

4.2.6 Βηματικοί κινητήρες

Για την λειτουργία των βηματικών κινητήρων δημιουργήθηκαν συναρτήσεις που επιτρέπουν την κίνηση του ρομπότ για συγκεκριμένο χρόνο με συγκεκριμένη ταχύτητα και βήματα.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι η προ εγκατεστημένη βιβλιοθήκη time και η RPi.GPIO. Στην συνέχεια, έγινε ο ορισμός και η ανάθεση των εξόδων του μικροεπεξεργαστή σύμφωνα με την συνδεσμολογία που περιγράφεται στον πίνακα (Πίνακας 9).

Οι συναρτήσεις forward() και backward() (Εικόνα 40) είναι υπεύθυνες για κίνηση του ρομπότ μπροστά και πίσω και δέχονται σαν ορίσματα την ταχύτητα, την απόσταση που θα διανύσει και την απόσταση που θα σταματήσει το ρομπότ αν εντοπίσει κάποιο εμπόδιο. Για λόγους ασφαλείας η ταχύτητα δεν μπορεί να είναι μεγαλύτερη από 10 μονάδες και η τιμή της απόστασης μικρότερη από πέντε εκατοστά.

Οι συναρτήσεις right() και left() (Εικόνα 41) ελέγχουν την στροφή του ρομπότ δεξιά και αριστερά. Δέχονται σαν ορίσματα την ταχύτητα, την απόσταση, την κατεύθυνση κίνηση και την απόσταση που θα σταματήσει το ρομπότ αν βρεθεί κάποιο εμπόδιο.

```
def forward(speed,dist,trust):
    gpio.output(DIR_m1,1) #direction
    gpio.output(DIR_m2,1)

    speed = speed*0.001
    if speed > 0.01: #max speed
        speed = 0.001

    if trust < 5:
        trust = 5

    for x in range(dist):
        if ultra.ultra_sonic_front(trust) or
        ir_front():
            gpio.output(STEP_m1,gpio.HIGH)
            gpio.output(STEP_m2,gpio.HIGH)
            sleep(speed)
            gpio.output(STEP_m1,gpio.LOW)
            gpio.output(STEP_m2,gpio.LOW)
            sleep(speed)

def backward(speed,dist,trust):
    gpio.output(DIR_m1,0) #direction
    gpio.output(DIR_m2,0)

    speed = speed*0.001
    if speed > 0.01: #max speed
        speed = 0.001

    if trust < 5:
        trust = 5

    for x in range(dist):
        if ultra.ultra_sonic_back(trust) or
        ir_back():
            gpio.output(STEP_m1,gpio.HIGH)
            gpio.output(STEP_m2,gpio.HIGH)
            sleep(speed)
            gpio.output(STEP_m1,gpio.LOW)
            gpio.output(STEP_m2,gpio.LOW)
            sleep(speed)
```

Εικόνα 40: Συναρτήσεις κίνησης του ρομπότ μπροστά/πίσω

```

def right(speed,dist,dir,trust):
    gpio.output(DIR_m1,dir) #direction
    speed = speed*0.001
    if speed > 0.01: #max speed
        speed = 0.001
    if trust < 5:
        trust = 5
    for x in range(dist):
        if dir == 1:
            if ultra.ultra_sonic_front(trust)
            or ir_front():
                gpio.output(STEP_m1,gpio.HIGH)
                sleep(speed)
                gpio.output(STEP_m1,gpio.LOW)
                sleep(speed)
            elif dir == 0:
                if ultra.ultra_sonic_back(trust)
                or ir_back():
                    gpio.output(STEP_m1,gpio.HIGH)
                    sleep(speed)
                    gpio.output(STEP_m1,gpio.LOW)
                    sleep(speed)

def left(speed,dist,dir,trust):
    gpio.output(DIR_m2,dir) #direction
    speed = speed*0.001
    if speed > 0.01: #max speed
        speed = 0.001
    if trust < 5:
        trust = 5
    for x in range(dist):
        if dir == 1:
            if ultra.ultra_sonic_front(trust)
            or ir_front():
                gpio.output(STEP_m2,gpio.HIGH)
                sleep(speed)
                gpio.output(STEP_m2,gpio.LOW)
                sleep(speed)
            elif dir == 0:
                if ultra.ultra_sonic_back(trust)
                or ir_back():
                    gpio.output(STEP_m2,gpio.HIGH)
                    sleep(speed)
                    gpio.output(STEP_m2,gpio.LOW)
                    sleep(speed)

```

Εικόνα 41: Συναρτήσεις κίνησης του ρομπότ δεξιά/αριστερά

4.3 Διασύνδεση με πληροφοριακό σύστημα

Το ρομπότ διαθέτει την δυνατότητα διασύνδεσης με πληροφοριακό σύστημα, από το οποίο ο χρήστης θα έχει την δυνατότητα να τρέχει τα δικά του σενάρια. Για να επιτευχθεί η ασφαλής διασύνδεση και οι δύο πλευρές είναι υποχρεωμένες να χρησιμοποιούν ένα εισιτήριο (token). Πρόκειται για ένα κομμάτι δεδομένων που δεν έχει κάποια χρήση μόνο του, αλλά σε συνδυασμό με το σωστό σύστημα αποκωδικοποίησης, αποτελεί ένα σημαντικό κομμάτι ασφάλειας. Το ρομπότ με την εκκίνηση του, ελέγχει εάν υπάρχει σύνδεση στο διαδίκτυο και στην συνέχεια εάν υπάρχει σύνδεση σε πληροφοριακό σύστημα. Μετά τον επιτυχή έλεγχο, το ρομπότ ζητάει ένα εισιτήριο (token), το οποίο χρησιμοποιεί για την μετέπειτα επικοινωνία με το σύστημα. Τα εισιτήρια για μεγαλύτερη ασφάλεια έχουν διάρκεια τριάντα λεπτά. Με την λήξη του εισιτηρίου το ρομπότ ζητάει εκ νέου εισιτήριο από το πληροφοριακό σύστημα.

```

def RobotCheckConnection(url,json_send):

    try:
        if requests.get('http://zafora.ece.uowm.gr/').ok:
            try:
                response = requests.post(url, json=json_send,timeout=.5)
                if response.status_code == 200:
                    print("Printing Entire Post Request")
                    print(response.json())
                    return response.json()

                elif response.status_code == 401:
                    print("Robot not connected with server")
                    return response.json()

            else:
                print("Unexpected status code")
                return False

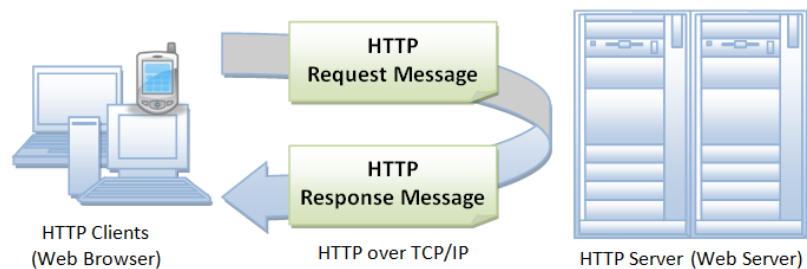
        except:
            print("No response")
            return False

    except:
        print("No Internet connection")
        return False

```

Εικόνα 42: Συνάρτηση RobotCheckConnection

Για την επικοινωνία του ρομποτικού συστήματος με το πληροφοριακό σύστημα και την παραλαβή εισιτηρίου (token) χρησιμοποιήθηκε συνάρτηση της βιβλιοθήκη requests. Συγκριμένα χρησιμοποιήθηκε η requests.post η οποία βασίζεται στο πρωτόκολλο HTTP (Hypertext Transfer Protocol). Το HTTP (Εικόνα 43) είναι ένα πρωτόκολλο που έχει σχεδιαστεί για να επιτρέπει την επικοινωνία μεταξύ πελατών και διακομιστών. Λειτουργεί ως πρωτόκολλο απόκρισης αιτήματος μεταξύ πελάτη και διακομιστή.



Εικόνα 43: Λειτουργία πρωτοκόλλου HTTP




Η συνάρτηση `RobotCheckConnection()` (**Εικόνα 42**) είναι υπεύθυνη για τον έλεγχο σύνδεσης του ρομπότ στο διαδίκτυο αλλά και για την διασύνδεση του με το πληροφοριακό σύστημα. Σαν πρώτο όρισμα δέχεται την διεύθυνση του πληροφοριακού συστήματος. Σαν δεύτερο όρισμα το όνομα και τον σειριακό αριθμό του ρομπότ σε αντικείμενο τύπου `json`. Στην περίπτωση ύπαρξης σύνδεσης στο διαδίκτυο, η συνάρτηση στέλνει το αντικείμενο τύπου `json` που περιέχει το όνομα και τον σειριακό αριθμό του ρομπότ. Αν η επιστρεφόμενη κατάσταση (`status`) είναι διακόσια, δηλαδή επιτυχής σύνδεση, παίρνει ως απάντηση ένα αντικείμενο που περιέχει το εισιτήριο (`token`), το οποίο θα χρησιμοποιηθεί στην συνέχεια για την επικοινωνία με το πληροφοριακό σύστημα. Σε περίπτωση που η κατάσταση ισοδυναμεί με τετρακόσια ένα, το όνομα και ο σειριακός αριθμός δεν είναι καταχωρημένα στο πληροφοριακό σύστημα, παίρνει ως απάντηση ένα αντικείμενο που περιέχει μήνυμα λάθους. Τέλος στην περίπτωση, που δεν υπάρχει σύνδεση με το πληροφοριακό ή σύνδεση στο διαδίκτυο επιστρέφει την τιμή `False`.

4.4 Γραφικά QtDesigner

Στην συγκεκριμένη ενότητα παρουσιάζεται και αναλύεται η δομή του γραφικού περιβάλλοντος χρήστη (GUI) του συστήματος. Με την βοήθεια του εργαλείου QtDesigner σχεδιάστηκαν και συνδέθηκαν τα επιμέρους αντικείμενα, τα οποία αποτελούν το γραφικό περιβάλλον του συστήματος. Η οθόνη αφής στην οποία εμφανίζονται τα γραφικά του συστήματος υποστηρίζει ανάλυση 800 pixels οριζόντια και 480 pixels κάθετα. Για αυτό το λόγο το παράθυρο της εφαρμογής το οποίο είναι ένα αντικείμενο `QMainWindow` έχει διαστάσεις 800x480 pixels.

Για την καλύτερη πλοήγηση του χρήστη δημιουργήθηκε μια γραμμή εργαλείων η οποία εμφανίζεται μόνιμα στην πάνω μεριά της οθόνης (**Εικόνα 44**). Στοιχισμένα στην δεξιά πλευρά βρίσκονται τρία αντικείμενα (**Εικόνα 45**) τύπου `QAction`. Το Πρώτο είναι το αντικείμενο `actionExit`, το οποίο είναι υπεύθυνο για την έξοδο από την εφαρμογή και τον τερματισμό του συστήματος. Μπορεί να χρησιμοποιηθεί με την συντόμευση `Ctrl+Q`. Δεύτερο στοιχείο είναι το αντικείμενο `actionHome`, το οποίο οδηγεί τον χρήστη στην αρχική διεπαφή. Χρησιμοποιείται και με την συντόμευση `Ctrl+H`. Τέλος δημιουργήθηκε το αντικείμενο `actionconnected` το οποίο παρουσιάζει την κατάσταση σύνδεσης του συστήματος με πληροφοριακό σύστημα. Το συγκεκριμένο αντικείμενο `QAction` εναλλάσσει τα χρώματα της εμφάνισης του ανάλογα με την κατάσταση σύνδεσης. Στην περίπτωση της επιτυχής σύνδεσης με πληροφοριακό σύστημα αποκτά

το χρώμα πράσινο σε αντίθετη περίπτωση γίνεται κόκκινο. Τα τρία αυτά αντικείμενα χωρίζονται μεταξύ τους με αντικείμενα διαχωριστές (separators).

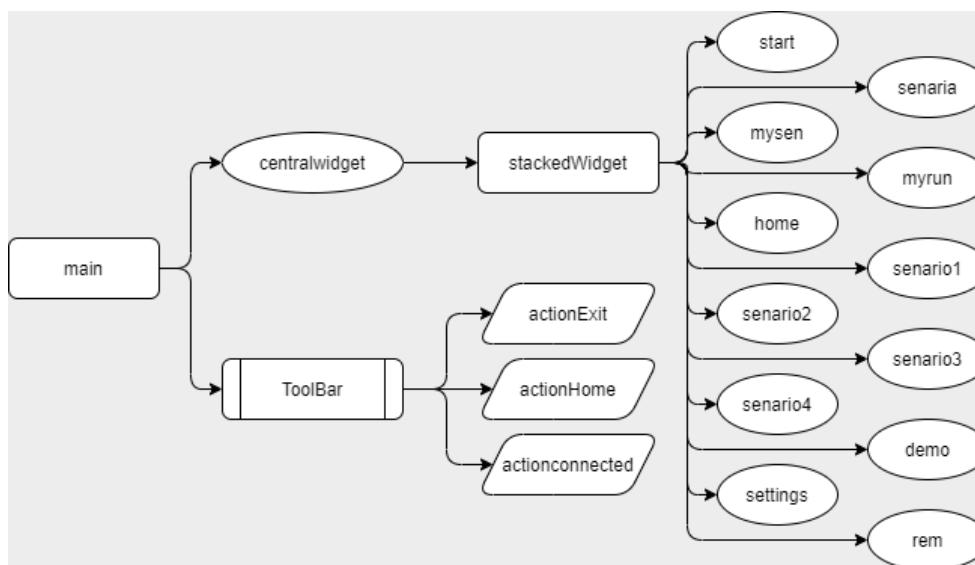
Name	Used	Text	Shortcut	Checkable	ToolTip
 actionExit	<input checked="" type="checkbox"/>	Exit	Ctrl+Q	<input type="checkbox"/>	Exit
 actionHome	<input checked="" type="checkbox"/>	Home	Ctrl+H	<input type="checkbox"/>	Home
 actionconnected	<input checked="" type="checkbox"/>	connected		<input type="checkbox"/>	connected

Εικόνα 44: Χαρακτηριστικά αντικειμένων QAction



Εικόνα 45: Στοιχεία γραμμής εργαλείων

Με διαστάσεις 800x430 pixels και μετατοπισμένο στον κατακόρυφο άξονα κατά 50 pixels χαμηλότερα δημιουργήθηκε ένα αντικείμενο τύπου QWidget, το οποίο είναι η κεντρική διεπαφή της εφαρμογής. Μέσα στην διεπαφή με ίδιες διαστάσεις τοποθετήθηκε ένα αντικείμενο στοιβαγμένων διεπαφών QStackedWidget (**Εικόνα 46**).

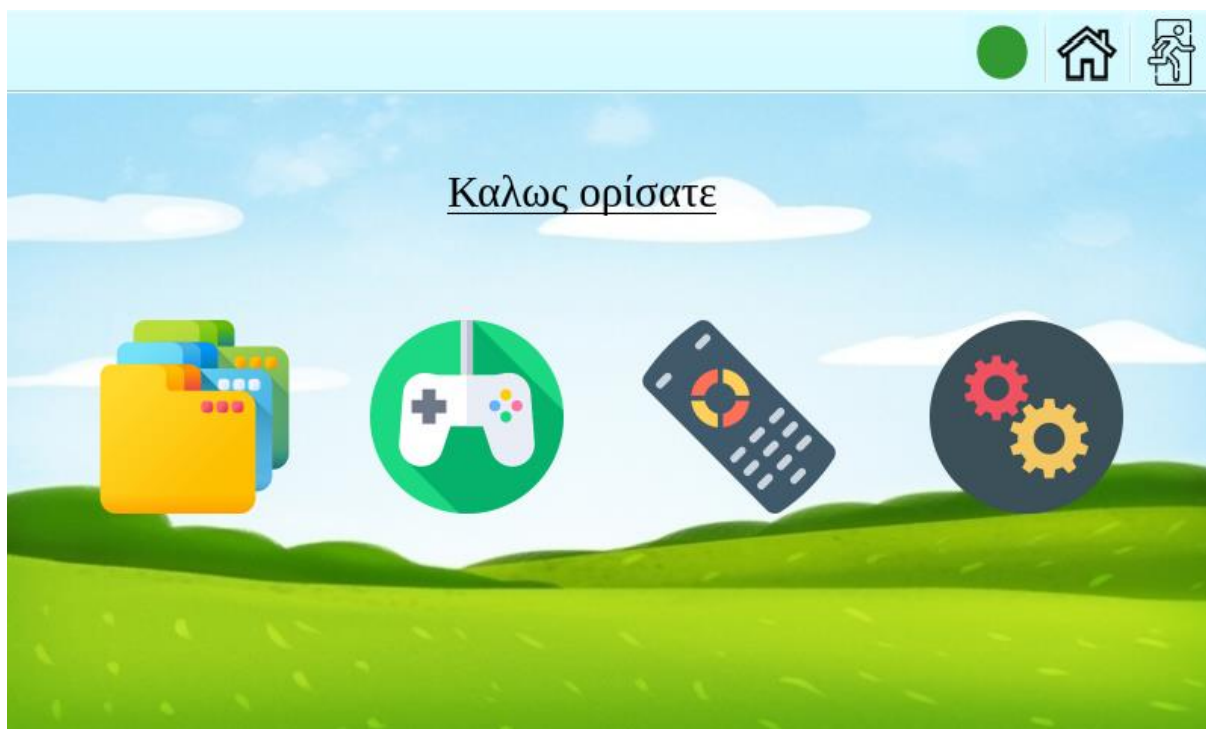


Εικόνα 46: Δομή διεπαφών και γραμμής εργαλείων

Παρακάτω παρουσιάζονται λεπτομερώς οι διεπαφές και τα αντικείμενα, τα οποία είναι εντός της διεπαφής stackedWidget. Αρχικά παρουσιάζεται η δομή (**Πίνακας 10**) και η εμφάνιση της διεπαφής start (**Εικόνα 47**)

Όνομα διεπαφής	start	
Όνομα αντικειμένου	Τύπος	Χρήση
label_start	QLabel	Τίτλος διεπαφής
senar_home	QPushButton	Οδηγεί στην διεπαφή κατηγοριών σεναρίων
games_home	QPushButton	Οδηγεί στην εφαρμογή των παιχνιδιών
remote_home	QPushButton	Οδηγεί στην διεπαφή ελέγχου τηλεχειρισμού
settings_home	QPushButton	Οδηγεί στην διεπαφή ρυθμίσεων δικτύου

Πίνακας 10: Διεπαφή αρχικής οθόνης



Εικόνα 47: Στιγμιότυπο διεπαφής αρχικής οθόνης

Έπειτα, παρουσιάζεται η δομή (**Πίνακας 11**) και η εμφάνιση της διεπαφής senaria για την εμφάνιση των κατηγοριών των σεναρίων (**Εικόνα 48**).

Όνομα διεπαφής	senaria	
Όνομα αντικειμένου	Τύπος	Χρήση
label_senaria	QLabel	Τίτλος διεπαφής
local_sen	QPushButton	Οδηγεί στην διεπαφή τοπικών σεναρίων
my_sen	QPushButton	Οδηγεί στην διεπαφή σεναρίων του χρήστη
return_senaria	QPushButton	Επιστρέφει στην διεπαφή start

Πίνακας 11: Διεπαφή επιλογής κατηγορίας σεναρίων



Εικόνα 48: Στιγμιότυπο διεπαφής επιλογής κατηγορίας σεναρίων

Η διεπαφή home η οποία περιέχει τα παρακάτω αντικείμενα (Πίνακας 12) παρουσιάζει τα προ εγκατεστημένα σενάρια (Εικόνα 49).

Όνομα διεπαφής	home	
Όνομα αντικειμένου	Τύπος	Χρήση
label_home	QLabel	Τίτλος διεπαφής
sen1	QPushButton	Οδηγεί στην διεπαφή του πρώτου σεναρίου
sen2	QPushButton	Οδηγεί στην διεπαφή του δεύτερου σεναρίου
sen3	QPushButton	Οδηγεί στην διεπαφή του τρίτου σεναρίου
sen4	QPushButton	Οδηγεί στην διεπαφή του τέταρτου σεναρίου
demo	QPushButton	Οδηγεί στην διεπαφή του σεναρίου επίδειξης
return_home	QPushButton	Επιστρέφει στην διεπαφή senaria

Πίνακας 12: Διεπαφή παρουσίασης προ εγκατεστημένων σεναρίων



Εικόνα 49: Στιγμιότυπο διεπαφής παρουσίασης προ εγκατεστημένων σεναρίων

Παρουσιάζεται η δομή των διεπαφών senario1 (Πίνακας 13), senario2 (Πίνακας 14), senario3 (Πίνακας 15), senario4 (Πίνακας 16), demo (Πίνακας 17), τα οποία βρίσκονται προ εγκατεστημένα στο ρομποτικό σύστημα.

Όνομα διεπαφής	senario1	
Όνομα αντικειμένου	Τύπος	Χρήση
cart_label_1	QLabel	Τίτλος διεπαφής
sen1_no	QPushButton	Επιδρά στην εναλλαγή των προσώπων
sen1_yes	QPushButton	Επιδρά στην εναλλαγή των προσώπων
faces_1	QStackedWidget	Περιέχει τις επιμέρους διεπαφές προσώπων

Πίνακας 13: Διεπαφή προ εγκατεστημένου σεναρίου «Ειδική Ενσυναίσθηση»

Όνομα διεπαφής	senario2	
Όνομα αντικειμένου	Τύπος	Χρήση
cart_label_2	QLabel	Τίτλος διεπαφής
sen2_no	QPushButton	Επιδρά στην εναλλαγή των προσώπων
sen2_yes	QPushButton	Επιδρά στην εναλλαγή των προσώπων
faces_2	QStackedWidget	Περιέχει τις επιμέρους διεπαφές προσώπων
img_2	QStackedWidget	Περιέχει τις επιμέρους διεπαφές ζώων

Πίνακας 14: Διεπαφή προ εγκατεστημένου σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»

Όνομα διεπαφής	senario3	
Όνομα αντικειμένου	Τύπος	Χρήση
cart_label_3	QLabel	Τίτλος διεπαφής
sen3_no	QPushButton	Επιδρά στην εναλλαγή των προσώπων
sen3_yes	QPushButton	Επιδρά στην εναλλαγή των προσώπων
faces_3	QStackedWidget	Περιέχει τις επιμέρους διεπαφές προσώπων

Πίνακας 15: Διεπαφή προ εγκατεστημένου σεναρίου «Εκφοβισμός»

Όνομα διεπαφής	senario4	
Όνομα αντικειμένου	Τύπος	Χρήση
cart_label_4	QLabel	Τίτλος διεπαφής
sen4_no	QPushButton	Επιδρά στην εναλλαγή των προσώπων
sen4_yes	QPushButton	Επιδρά στην εναλλαγή των προσώπων
faces_4	QStackedWidget	Περιέχει τις επιμέρους διεπαφές προσώπων

Πίνακας 16: Διεπαφή προ εγκατεστημένου σεναρίου «Κοινωνική Συμπεριφορά»

Όνομα διεπαφής	demo	
Όνομα αντικειμένου	Τύπος	Χρήση
cart_label_5	QLabel	Τίτλος διεπαφής
img	QLabel	Πεδίο προβολής φωτογραφίας
photo	QPushButton	Έναρξη λειτουργίας φωτογράφισης
play	QPushButton	Έναρξη λειτουργίας επίδειξης

Πίνακας 17: Διεπαφή προ εγκατεστημένου σεναρίου «Επίδειξης»

Η διεπαφή settings (Πίνακας 18) είναι υπεύθυνη για την παρουσίαση βασικών στοιχείων του ρομπότ, όπως το όνομα του, καθώς και των στοιχείων σύνδεσης στο διαδίκτυο (Εικόνα 50).

Όνομα διεπαφής	settings	
Όνομα αντικειμένου	Τύπος	Χρήση
label_settings	QLabel	Τίτλος διεπαφής
label_name	QLabel	Τίτλος
input_name	QLabel	Όνομα ρομπότ
label_ip	QLabel	Τίτλος
input_ip	QLabel	Διεύθυνση ip του ρομπότ

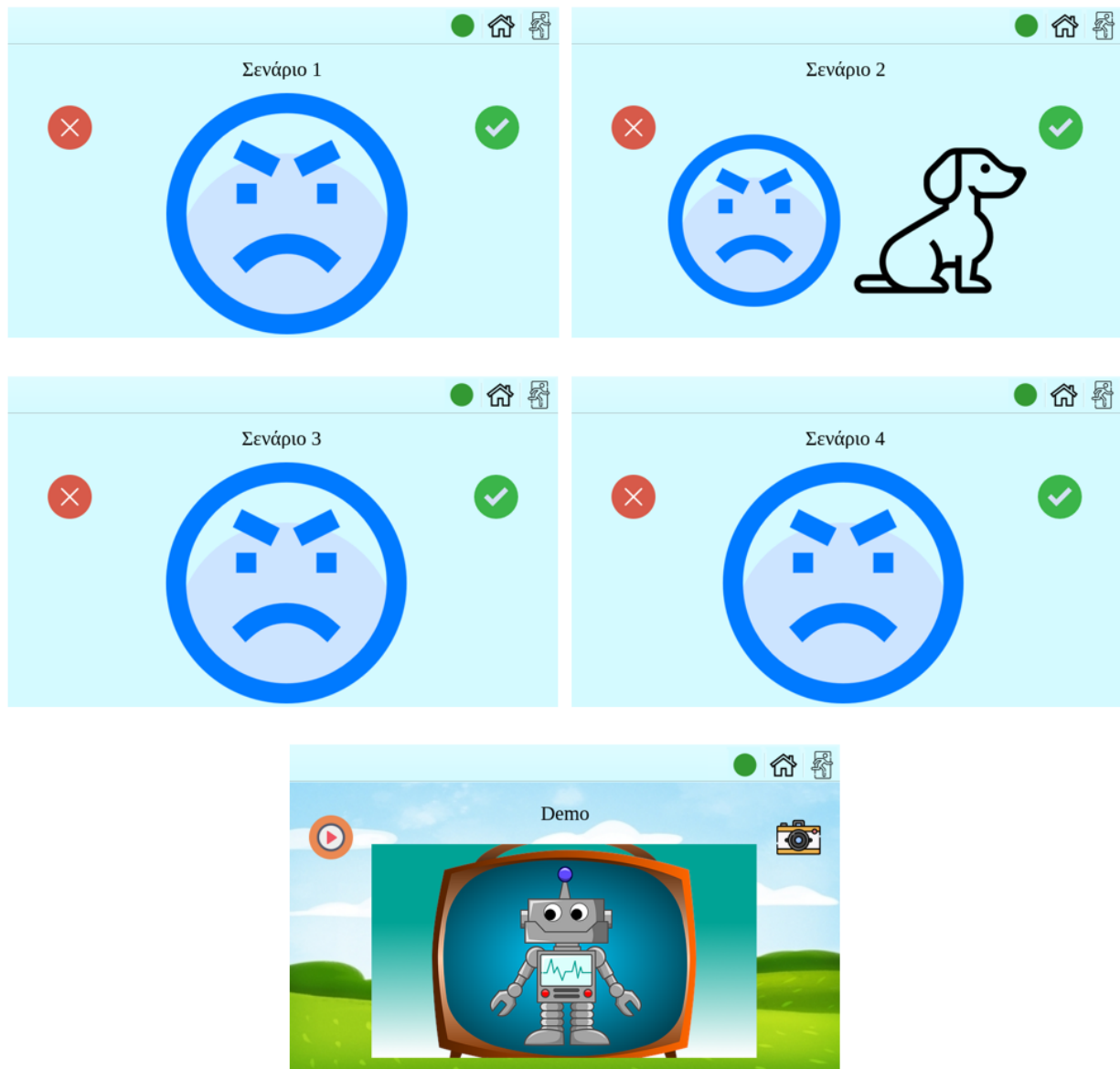
label_gateway	QLabel	Τίτλος
input_gate	QLabel	Το Gateway του δικτύου
label_public	QLabel	Τίτλος
input_public	QLabel	Διεύθυνση δημόσιας ip
label_network	QLabel	Τίτλος
line	Line	Διαχωριστικό διεπαφής
line_2	Line	Διαχωριστικό διεπαφής
line_3	Line	Διαχωριστικό διεπαφής
connection	QPushButton	Οδηγεί σε μήνυμα πληροφόρησης
return_home	QPushButton	Επιστρέφει στην διεπαφή start

Πίνακας 18: Διεπαφή ρυθμίσεων δικτύου



Εικόνα 50: Στιγμιότυπο διεπαφής ρυθμίσεων δικτύου

Παρατίθενται οι εμφανίσεις των πέντε προ εγκατεστημένων σεναρίων (Εικόνα 51)



Εικόνα 51: Στιγμιότυπα διεπαφών προ εγκατεστημένων σεναρίων

Οι στοιβαγμένες διεπαφές $faces_k$ ($k = \{1, 2, 3, 4\}$) περιέχουν διεπαφές οι οποίες περιέχουν τα αντίστοιχα αντικείμενα όπως απεικονίζεται στον πίνακα (Πίνακας 19).

Όνομα στιβαγμένης διεπαφής	faces_k	
Όνομα εσωτερικής διεπαφής	Όνομα αντικειμένου	Τύπος
face1_n	face1_label1_n	QLabel
face2_n	face2_label2_n	QLabel
face3_n	face3_label3_n	QLabel
face4_n	face4_label4_n	QLabel
face5_n	face5_label5_n	QLabel

Πίνακας 19: Εσωτερική διεπαφή απεικόνισης εκφράσεων προσώπου

Η στιβαγμένη διεπαφή *img_2* περιλαμβάνει τις εσωτερικές διεπαφές *img1_2*, *img2_2*, *img3_2* οι οποίες περιέχουν αντικείμενα όπως εμφανίζεται στον πίνακα (Πίνακας 20).

Όνομα στιβαγμένης διεπαφής	img_2	
Όνομα εσωτερικής διεπαφής	Όνομα αντικειμένου	Τύπος
img1_2	img1_label1_2	QLabel
img2_2	img2_label2_2	QLabel
img3_2	img3_label3_2	QLabel

Πίνακας 20: Εσωτερική διεπαφή απεικόνισης ζώων

4.5 Βασική συνάρτηση PyQt v5

Στην ενότητα αυτή περιγράφεται το εκτελέσιμο αρχείο του ρομποτικού συστήματος. Ο μικροεπεξεργαστής του ρομπότ ρυθμίστηκε ώστε με την εκκίνηση του να εκτελεί το βασικό αρχείο python από το οποίο ελέγχονται όλες οι επιμέρους κλάσεις που έχουν αναφερθεί στις προηγούμενες ενότητες.

Με την έναρξη του ρομπότ, το αρχείο ui που περιέχει τα αντικείμενα των γραφικών εκτελείται με την συνάρτηση `load_ui()` (**Εικόνα 52**), η οποία συνδέει τα αντικείμενα με τις συναρτήσεις τους.

```
class main(QMainWindow):
    def __init__(self):
        super(main, self).__init__()
        self.load_ui()

    def load_ui(self):
        ui_file = (Path(__file__).parent / "form.ui").resolve()
        loadUi(ui_file, self)
        self.showFullScreen()

        led.set_led(0, 0, 255)
        wave_obj = sa.WaveObject.from_wave_file("sounds/home.wav") #sound start
        play_obj = wave_obj.play()

        self.actionExit.triggered.connect(self.quit) #Exit button
        self.actionHome.triggered.connect(self.homes) #Home button

        self.games_home.clicked.connect(self.games) #Games
        self.senar_home.clicked.connect(self.senarios) #Scenarios list
        self.settings_home.clicked.connect(self.settings) #settings
        self.remote_home.clicked.connect(self.remote) #Remote contol

        self.local_sen.clicked.connect(self.loc_sen)
        self.my_sen.clicked.connect(self.mys)

        self.sen1.clicked.connect(self.s1)
        self.sen2.clicked.connect(self.s2)
        self.sen3.clicked.connect(self.s3)
        self.sen4.clicked.connect(self.s4)
        self.dem.clicked.connect(self.demos)

        self.return_senaria.clicked.connect(self.r_start) #Back button
        self.return_settings.clicked.connect(self.r_start)

        self.return_home.clicked.connect(self.r_senaria)
        self.return_mysen.clicked.connect(self.r_senaria)
```

Εικόνα 52: Δήλωση γραφικών στην βασική συνάρτηση

Στην **Εικόνα 53** και στην **Εικόνα 54** παρουσιάζονται οι συναρτήσεις που είναι υπεύθυνες για το έλεγχο λειτουργίας των αντικειμένων της γραμμής εργαλείων (toolbar).

Η συνάρτηση `server_connection()` αναλαμβάνει την παράλληλη εκτέλεση του αρχείου `server_connection.py` του οποίου η λειτουργία έχει περιγραφεί στην **ενότητα 4.3**. Με την συνάρτηση `con_server()` γίνεται η ενημέρωση της μεταβλητής που είναι υπεύθυνη για την κατάσταση σύνδεση.

Η συνάρτηση `change_icon_con()` είναι υπεύθυνη για την εμφάνιση της κατάστασης σύνδεσης (`actionconnected`) στην γραμμή εργαλείων (`toolbar`). Στην περίπτωση που η μεταβλητή `server` γίνει αληθής, αλλάζει το χρώμα του αντικειμένου σε πράσινο, που σημαίνει ότι υπάρχει διαθέσιμο πληροφοριακό σύστημα και έχει επιτευχθεί με επιτυχία η διασύνδεση του ρομπότ με αυτό. Σε αντίθετη περίπτωση θέτει το χρώμα κόκκινο που υποδηλώνει την απώλεια πληροφοριακού συστήματος ή αδυναμία σύνδεσης σε αυτό.

Η συνάρτηση `find_commands.find_manual()` είναι υπεύθυνη για τον σωστό τερματισμό των αρχείων που τρέχουν παράλληλα με το βασικό αρχείο.

Με το πάτημα του κουμπιού `actionHome` εκτελείται η συνάρτηση `home()` η οποία επιστρέφει τον χρήστη στην αρχική σελίδα του συστήματος.

Με το πάτημα του κουμπιού `actionExit` εκτελείται η συνάρτηση `quit()` η οποία τερματίζει όλα τα αρχεία που τρέχουν παράλληλα με το ρομπότ και στην συνέχεια κλείνει το σύστημα.

```
def server_connection():
    print("Network Connectio: Take new token from server ")
    find_commands.find_kill_manual('server_connection.py')
    os.system("sudo python3 server_connection.py")
def change_icon_con(self, widget): #change red/green icon for connection
    global server
    if server:
        self.actionconnected.setIcon(QIcon("images/green_dot.png"))
    else:
        self.actionconnected.setIcon(QIcon("images/red_dot.png"))
def con_server(): #True(green_dot)/False(red_dot) and split token
    global server, widget, t
    flag = True #First time
    cur_time = time.time()
    while True:
        sleep(1)
        c_time = time.time()
        if c_time-cur_time >= 1800 or flag:
            try:
                f = open("network_check.txt", "r")
                ser = f.read()
                s, t = ser.split('*')
                if str(s) == "1":
                    server = True
                else:
                    server = False
            except:
                server = False
            widget.change_icon_con(widget)
            cur_time = time.time()
            flag = False
if stop_threads:
    os.system("pkill -f server_connection.py")
    f.close()
    break
```

Εικόνα 53: Συναρτήσεις ενημέρωσης κατάστασης σύνδεσης


```

def homes(self):
    global run_sen4
    sa.stop_all() #sound
    widget.stackedWidget.setCurrentWidget(widget.start)

def quit(self):
    global app, stop_threads
    sa.stop_all() #sound
    wave_obj = sa.WaveObject.from_wave_file("sounds/close.wav") #sound close
    play_obj = wave_obj.play()
    app.quit()
    stop_threads = True

```

Εικόνα 54: Συναρτήσεις γραμμής εργαλείων

Οι συνάρτησης `senarios()`, `remote()`, `settings()` (**Εικόνα 55**) είναι υπεύθυνες για την αλλαγή των διεπαφών (`widget`). Η συνάρτηση `games()` εκτελεί την εφαρμογή `gcompris` που περιέχει τα εκπαιδευτικά παιχνίδια του συστήματος.

```

def senarios(self):
    widget.stackedWidget.setCurrentWidget(widget.senaria)
def games(self):
    QApplication.setOverrideCursor(Qt.WaitCursor)
    os.system('cd /usr/games && ./gcompris-qt')
    QApplication.restoreOverrideCursor()
def remote(self):
    widget.stackedWidget.setCurrentWidget(widget.rem)
def settings(self):
    QApplication.setOverrideCursor(Qt.WaitCursor)
    settings.settings.load_setui(self, widget)
    QApplication.restoreOverrideCursor()

```

Εικόνα 55: Συναρτήσεις εμφάνισης και εκτέλεσης των αντικείμενων αρχικής οθόνης

Για την εμφάνιση των ρυθμίσεων είναι υπεύθυνη η κλάση `settings()` (**Εικόνα 57**). Η συνάρτηση `load_setui()` εκτελεί τα γραφικά και το αρχείο `python network.py` (**Εικόνα 56**), το οποίο ελέγχει αν έχει συνδεθεί το ρομπότ στο δίκτυο με την αναζήτηση της διεύθυνση IPv4 ενός τομέα (`domain name`). Στην συνέχεια η `get_ip_address()` επιστρέφει την τοπική

διεύθυνση (local ip), η `get_default_gateway_linux()` επιστέφει την διεύθυνση του δρομολογητή (gateway) και η `is_connected()` την δημόσια διεύθυνση (public ip). Η συνάρτηση `message_con()` εμφανίζει στον χρήστη της οδηγίες για την σύνδεση του ρομπότ στο δίκτυο μέσω του εργαλείου `comitup`. Το `comitup` θέτει την κάρτα δικτύου του μικροεπεξεργαστή μας σε σταθμό βάσης (access point). Με την χρήση άλλης συσκευής συνδεόμαστε στο ελεύθερο δίκτυο που έχει δημιουργήσει. Στην σελίδα που εμφανίζεται επιλέγουμε και εισάγουμε τον κωδικό πρόσβασης του δικτύου που θέλουμε να συνδεθεί το ρομπότ. Μετά από μερικά λεπτά η κατάσταση της κάρτας δικτύου επαναφέρεται στην αρχική κατάσταση και συνδέεται στο επιλεγμένο δίκτυο.

```
def is_connected(REMOTE_SERVER):    #Find Public IP address
    try:
        host = socket.gethostbyname(REMOTE_SERVER)
        s = socket.create_connection((host, 80), 2)
        s.close()
        return True
    except:
        pass
    return False

def get_ip_address(ifname):        #Local ip address
    s = 0
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        return socket.inet_ntoa(fcntl.ioctl(
            s.fileno(),
            0x8915, # SIOCGIFADDR
            struct.pack('256s', ifname[:15])
        )[20:24])
    except:
        return False

def get_default_gateway_linux():   #Gateway ip address
    with open("/proc/net/route") as fh:
        for line in fh:
            fields = line.strip().split()
            if fields[1] != '00000000' or not int(fields[3], 16) & 2:
                continue
            try:
                return socket.inet_ntoa(struct.pack("<L", int(fields[2], 16)))
            except:
                return False
```

Εικόνα 56: Συναρτήσεις εύρεσης διευθύνσεων δικτύου

```

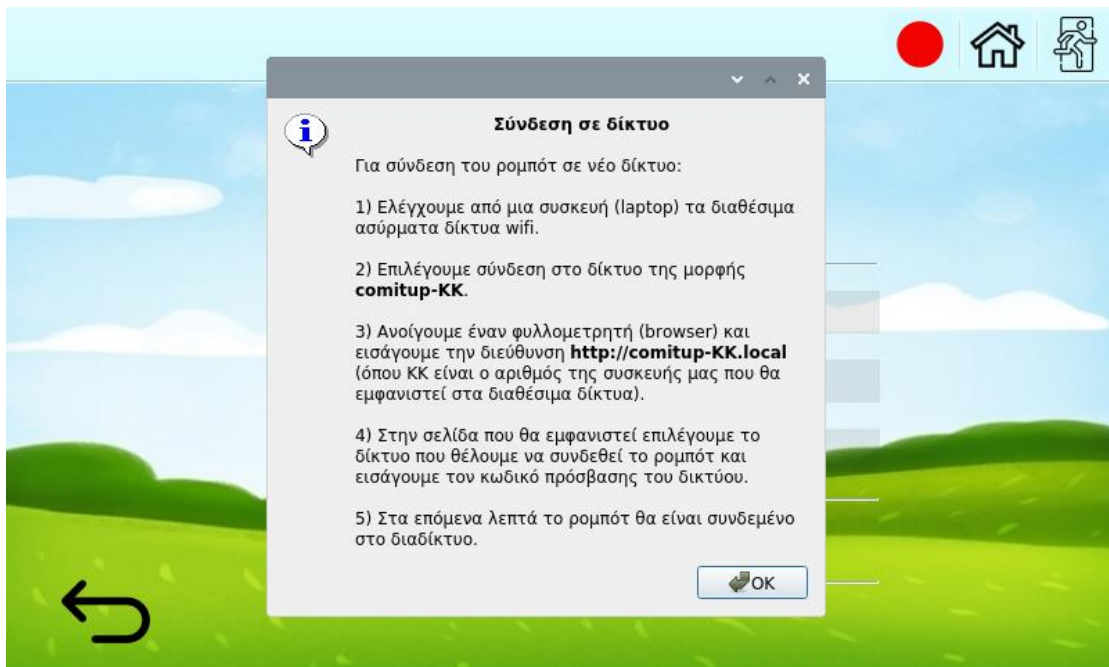
class settings(QWidget):
    def load_setui(self, widget):
        global flag
        if (flag):
            self.connection.clicked.connect(partial(settings.message_con, self))
            flag = False
        widget.stackedWidget.setCurrentWidget(widget.settings)
        proc = subprocess.Popen(["python network_python.py"], stdout=subprocess.PIPE, shell=True)
        (out) = proc.communicate()
        o = str(out)
        a, net_ip, b, gate, c, public, d = o.split('+')
        self.input_ip.setText(net_ip)
        self.input_gate.setText(gate)
        self.input_public.setText(public)

    def message_con(self):
        self.msg = QMessageBox()
        self.msg.setText("<center><b>Σύνδεση σε δίκτυο</b></center> <br> Για σύνδεση του ρομπότ σε νέο δίκτυο: <br><br> 1) Ελέγχουμε από μια συσκευή (laptop) τα διαθέσιμα ασύρματα δίκτυα wifi. <br><br> 2) Επιλέγουμε σύνδεση στο δίκτυο της μορφής <b>comitup-KK</b>. <br><br> 3) Ανοίγουμε έναν φυλλομετρητή (browser) και εισάγουμε την διεύθυνση <b>http://comitup-KK.local</b> (όπου KK είναι ο αριθμός της συσκευής μας που θα εμφανιστεί στα διαθέσιμα δίκτυα). <br><br> 4) Στην σελίδα που θα εμφανιστεί επιλέγουμε το δίκτυο που θέλουμε να συνδεθεί το ρομπότ και εισάγουμε τον κωδικό πρόσβασης του δικτύου. <br><br> 5) Στα επόμενα λεπτά το ρομπότ θα είναι συνδεδεμένο στο διαδίκτυο.")
        self.msg.setIcon(QMessageBox.Information)
        self.msg.show()

```

Εικόνα 57: Κλάση ρυθμίσεων δικτύου του συστήματος

Με το πάτημα του κουμπιού «σύνδεση» εμφανίζονται οι οδηγίες σύνδεσης του ρομπότ στο διαδίκτυο (Εικόνα 58).



Εικόνα 58: Οδηγίες σύνδεσης στο διαδίκτυο

Στην Εικόνα 59 παρουσιάζονται οι συναρτήσεις, που είναι υπεύθυνες τόσο για τα προ εγκατεστημένα σενάρια, όσο και για τα σενάρια που δημιουργεί ο χρήστης.

Η συνάρτηση loc_sen() οδηγεί στην διεπαφή home στην οποία βρίσκονται τα προ εγκατεστημένα σενάρια.

Η συνάρτηση `mys()` είναι υπεύθυνη για την λήψη των σεναρίων του χρήστη από το πληροφοριακό σύστημα. Για την παραλαβή του συνδέσμου λήψης είναι απαραίτητη η αποστολή του εισιτηρίου (token). Η αποστολή γίνεται με `get.requests()` στον server και στα πεδία του περιέχεται το εισιτήριο. Μετά την επιτυχή τακτοποίηση από το πληροφοριακό στην απάντηση του server περιέχεται ο σύνδεσμος λήψης του συμπιεσμένου αρχείου. Γίνεται λήψη, αποσυμπίεση και κλήση της συνάρτησης `my_files()` η οποία εκτελεί το αρχείο του χρήστη.

```

def loc_sen(self):
    widget.stackedWidget.setCurrentWidget(widget.home)

def my_files(self, Run_file): #run user files
    print (Run_file)
    widget.stackedWidget.setCurrentWidget(widget.myrun)
    self.label_myrun.setText(Run_file)
    Run_file, a = Run_file.split('.')
    mymodule = importlib.import_module(Run_file)
    mymodule.my_program.start(self, widget)

def mys(self): #Connection with server, download and extract files [Send token and take download link]
    global t
    QApplication.setOverrideCursor(Qt.WaitCursor)
    widget.stackedWidget.setCurrentWidget(widget.mysen)
    try:
        if requests.get('http://zafora.ece.uowm.gr').ok: #check internet connection
            try:
                download_link = "/no url"
                a, b, c, my_t, d = t.split("") #split token from json
                myUrl = 'http://zafora.ece.uowm.gr:8080/scene' #server link
                head = {'Authorization': 'Bearer ' + my_t} #header with token
                resp = requests.get(myUrl, headers=head) #get requests
                download_link = resp.json() #return download link
                print("Pending : " + str(resp.status_code))
                print(my_t)
                print(download_link)
                wget.download(download_link, '/home/pi/Desktop/project/my_senario/')
            except:
                print("I cant split token in [''] or problem with json and status_code")

                a = download_link.split("/") #split file name from download link
                b = a[len(a)-1] #file name with type
                c = "my_senario/"+b
                try:
                    with ZipFile(c, 'r') as zipObj: #Extract zip file
                        zipObj.extractall("my_senario")
                    my_files(b)
                except:
                    print("I cant find file or unzip file")
            except:
                print("No internet")
    QApplication.restoreOverrideCursor()

```

Εικόνα 59: Συναρτήσεις εκτέλεσης προ εγκατεστημένων και σεναρίων χρήστη

Οι συναρτήσεις `return_start()`, `return_senaria()` (Εικόνα 60) με το πάτημα των αντικειμένων `return_home` και `return_senaria` επιστέφουν τον χρήστη στην προηγούμενη διεπαφή.

Με το πάτημα του κουμπιού πολλαπλών χρήσεων γίνεται η εκτέλεση της συνάρτησης `funstop()`, η οποία ανάλογα με την διάρκεια πατήματος του κουμπιού επιστρέφει στην αρχική οθόνη ή τερματίζει το ρομποτικό σύστημα όπως αναφέρθηκε στην **παράγραφο 4.2.1**. Κατά τον τερματισμό του συστήματος γίνεται η κλήση της συνάρτησης `killpy()`, η οποία διαβάζει από το αρχείο `open_pyhton.txt` τα αρχεία `pyhton` που εκτελούνται στο σύστημα και τα τερματίζει.

```

def return_start(self):
    sa.stop_all() #sound
    widget.stackedWidget.setCurrentWidget(widget.start)
def return_senaria(self):
    sa.stop_all() #sound
    widget.stackedWidget.setCurrentWidget(widget.senaria)

def killpy(self): #kill all prosses
    f = open("open_pyhton.txt", "r")
    for x in f:
        os.system("pkill -f " + str(x))
def funstop():
    print("function stop button")
    button = Button(16)
    end = 0
    while(1):
        sleep(1)
        start = time.time()
        while button.is_pressed: #if button pressed
            end = time.time() #count seconds
            print(end-start)
            if end - start >= 2: #return home page
                widget.killpy()
                #Return back
                #index(0)->start, index(1)->senaria, index(4)->home
                cur = widget.stackedWidget.currentIndex()
                if cur == 1 or cur == 10 or cur == 11 or cur == 12:
                    start = time.time()
                    widget.stackedWidget.setCurrentIndex(0)
                elif cur == 2 or cur == 3 or cur == 4:
                    start = time.time()
                    widget.stackedWidget.setCurrentIndex(1)
                elif cur == 5 or cur == 6 or cur == 7 or cur == 8 or cur == 9:
                    start = time.time()
                    widget.stackedWidget.setCurrentIndex(4)
            if end - start >= 8: #power off robot system
                widget.quit()
                break
    if stop_threads:
        sa.stop_all() #stop all sound
        break

```

Εικόνα 60: Συναρτήσεις επιστροφής και τερματισμού

Οι συναρτήσεις `s1()`, `s2()`, `s3()`, `s4()` και `demo()` (**Εικόνα 61**) εκτελούν τα αρχεία των προ εγκατεστημένων σεναρίων.

```

def s1(self):
    find_commands.find_kill("senario_1.py")
    senario_1.sen1.start(self, widget)
def s2(self):
    find_commands.find_kill("senario_2.py")
    senario_2.sen2(widget)
def s3(self):
    find_commands.find_kill("senario_3.py")
    senario_3.sen3.start(self, widget)
def s4(self):
    global run_sen4
    find_commands.find_kill("senario_4.py")
    senario_4.sen4.start(self, widget)
    run_sen4 = True
def demos(self):
    find_commands.find_kill("demo.py")
    widget.stackedWidget.setCurrentWidget(widget.demo)
    demo.demo(widget)

```

Εικόνα 61: Συναρτήσεις εκτελέσεις προ εγκατεστημένων σεναρίων

4.6 Προ εγκατεστημένα σενάκια

Σε συνεργασία με το Παιδαγωγικό Τμήμα Νηπιαγωγών του Πανεπιστημίου Δυτικής Μακεδονίας και την ερευνητική ομάδα του καθηγητή κύριου Μπράτιση Θαρρενού, μέλος του εργαστηρίου Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων, μελετήθηκαν και αναπτύχθηκαν τα σενάκια στα πλαίσια της συναισθηματικής νοημοσύνης τα οποία διαθέτει το ρομπότ. Τα σενάκια αυτά εστιάζουν στην συμπεριφορά ατόμων με διαταραχές στο αυτιστικό φάσμα, προκειμένου να προσομοιωθούν και υιοθετηθούν από το ρομποτικό σύστημα. Παρακάτω αναφέρονται και αναλύονται η δομή και τα περιεχόμενα των προ εγκατεστημένων σεναρίων που διαθέτει το ρομπότ.

Προκειμένου να γίνουν οι αντιδράσεις του ρομπότ όσο το δυνατόν κατανοητές και ξεκάθαρες προς τα παιδιά μικρής ηλικίας, σχεδιάστηκαν γραφικά στην οθόνη, που διαθέτει το ρομπότ στο κεφάλι, μια εικόνα προσώπου σε απλή μορφή τύπου emoji (**Εικόνα 62**). Υπάρχει δυνατότητα έκφρασης πέντε διαφορετικών καταστάσεων συναισθημάτων, η πρώτη κατάσταση υποδηλώνει τον έντονο εκνευρισμό που μπορεί να εκφράσει το ρομπότ ενώ η πέμπτη κατάσταση παρουσιάζει την έντονη χαρά του ρομπότ. Οι ενδιάμεσες καταστάσεις (δεύτερη, τρίτη, τέταρτη)

εκφράζουν προοδευτικά τα ενδιάμεσα συναισθήματα που υπάρχουν ανάμεσα στον εκνευρισμό και την χαρά.



Εικόνα 62: Η προσομοίωση των πέντε συναισθηματικών καταστάσεων

Στα πλαίσια της ολοκληρωμένης και πληρέστερης διαχείρισης του ρομπότ τοποθετήθηκαν αριστερά και δεξιά από το πρόσωπο δύο κουμπιά χειρισμού διάθεσης (**Εικόνα 63**) του ρομπότ. Το πρώτο απεικονίζει ένα κόκκινο στρογγυλό X, το οποίο όταν πατηθεί θέτει αυτομάτως το ρομπότ στην πιο ευδιάθετη και χαρούμενη του κατάσταση. Το δεύτερο απεικονίζει ένα πράσινο στρογγυλό ✓, το οποίο στο πάτημα του θέτει το πρόσωπο του ρομπότ κατά μια καλύτερη κατάσταση από την υπάρχουσα. Η χρήση των κουμπιών συνοδεύεται με έναν ήχο που υποδηλώνει την αλλαγή διάθεσης του ρομπότ.



Εικόνα 63: Κουμπιά αλλαγής διάθεσης του ρομπότ

4.6.1 Σενάριο «Ειδική Ενσυναίσθηση»

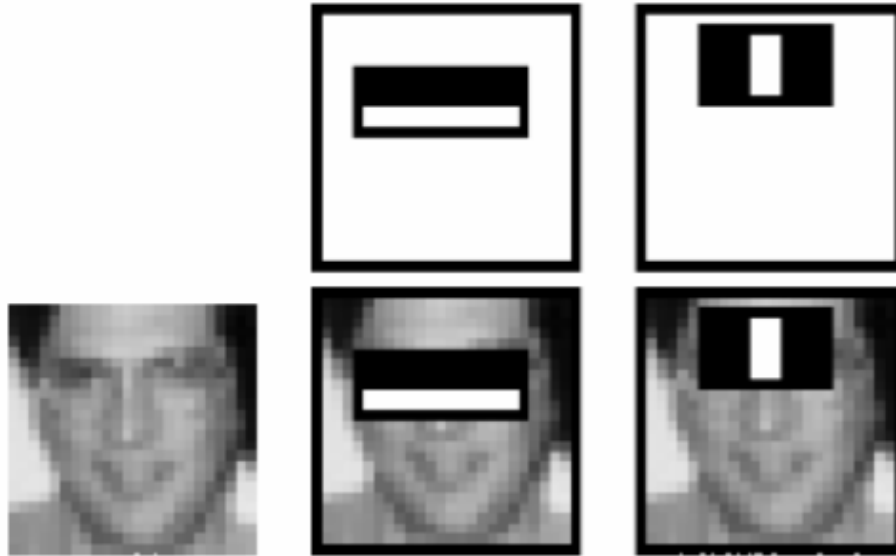
Το συγκεκριμένο σενάριο έχει ως στόχο να φέρει σε πρώτη επαφή τα παιδιά τυπικής ανάπτυξης με άτομα, τα οποία αντιμετωπίζουν διαταραχές στο φάσμα του αυτισμού. Σκοπός του είναι να υπάρξει κατανόηση της συμπεριφοράς που παρουσιάζουν τα άτομα με τις συγκεκριμένες διαταραχές μέσω της κοινωνικής αλληλεπίδρασης. Βασικό μέρος αποτελεί η ανάπτυξη της ενσυναίσθησης των παιδιών.

Το σενάριο χωρίζεται σε τρεις φάσεις, στην πρώτη ο εκπαιδευτικός παρουσιάζει υλικό που αφορά άτομα με ειδικές εκπαιδευτικές ανάγκες. Στην συνέχεια αναπτύσσεται μια συζήτηση περί του θέματος. Στην δεύτερη φάση ο εκπαιδευτικός και τα παιδιά αλληλεπιδρούν με το ρομπότ και

αυτό εκφράζει τις αντιδράσεις του (χαρά, θυμό, λύπη). Τέλος στην τρίτη φάση διεκπεραιώνεται η ανασκόπηση της συμπεριφοράς του ρομπότ και κατ' επέκταση της συμπεριφοράς των ατόμων με αυτές τις διαταραχές. Παρακάτω αναλύεται λεπτομερώς η δεύτερη φάση, στην οποία αναλύθηκαν, κωδικοποιήθηκαν και υλοποιήθηκαν οι συμπεριφορές των ατόμων με ειδικές εκπαιδευτικές ανάγκες.

Τα παιδιά με διαταραχές αυτιστικού φάσματος παρουσιάζουν συγκεκριμένες στερεοτυπικές συμπεριφορές, στο ρομπότ προστέθηκαν οι πιο έντονες από αυτές. Η πρώτη αντίδραση που δημιουργεί το ρομπότ προκύπτει από την δράση του χρήστη να το κοιτάζει στα μάτια. Κατά την συγκεκριμένη δράση το ρομπότ κατεβάζει το κεφάλι, αλλάζει την εικόνα του προσώπου του κατά μια κατάσταση πιο θυμωμένο, αλλάζει το χρώμα του LED, που υπάρχει στο στήθος του σε κόκκινο, ακούγεται ένας ήχος απογοήτευσης και σηκώνει τα χέρια του. Η δεύτερη αντίδραση που έχει το ρομπότ είναι στην περίπτωση που το πλησιάσει κάποιος αυτό προσπαθεί να κρατήσει την αρχική απόσταση κάνοντας όπισθεν, επίσης αλλάζει και σε αυτή την περίπτωση το χρώμα του LED σε κόκκινο και ακούγεται ο αντίστοιχος ήχος. Καθ' όλη την διάρκεια εκτέλεσης του σεναρίου εκτελείται ένα συγκεκριμένο κινηματικό μοντέλο, με το οποίο εκφράζει μια εμφανή στερεοτυπική συμπεριφορά.

Ειδικότερα στην πρώτη αντίδραση το ρομπότ αντιλαμβάνεται το βλέμμα του χρήστη μέσω της κάμερας που διαθέτει. Με την χρήση των βιβλιοθηκών *cv2* (OpenCV), *numpy* και *os* (λειτουργικού συστήματος) αναπτύχθηκε ένας κώδικας εντοπισμού προσώπου (face detection). Από την βιβλιοθήκη *cv2* επιλέχθηκε η κλάση *CascadeClassifier* με την οποία εντοπίζονται οι αλληλουχίες του προσώπου και των ματιών. Πιο συγκεκριμένα, η κλάση *CascadeClassifier* ανιχνεύει αντικείμενα που χρησιμοποιούν ταξινομητές Cascade βάσει χαρακτηριστικών Haar είναι μια αποτελεσματική μέθοδος ανίχνευσης αντικειμένων. Πρόκειται για μια προσέγγιση βασισμένη στη μηχανική μάθηση όπου η λειτουργία Cascade εκπαιδεύεται από πολλές θετικές και αρνητικές εικόνες. Στη συνέχεια χρησιμοποιείται για την ανίχνευση αντικειμένων σε άλλες εικόνες. Στην περίπτωση ανίχνευσης προσώπου ο αλγόριθμος χρειάζεται πολλές θετικές εικόνες (εικόνες προσώπων) και αρνητικές εικόνες (εικόνες χωρίς πρόσωπα) για να εκπαιδεύσει τον ταξινομητή. Με την συγκεκριμένη μέθοδο συγκρίνονται ομάδες κοντινών εικονοστιχείων μεταξύ τους και εξάγει σε αποτέλεσμα μοτίβα λευκού και μαύρου (**Εικόνα 64**).



Εικόνα 64: Παράδειγμα εντοπισμού προσώπου

Η συνάρτηση `face_detection()` (Εικόνα 65) εντοπίζει μέσω της Pi camera και του OpenCV τα πρόσωπα που το κοιτάνε το ρομπότ. Η συνάρτηση επιστρέφει μια αληθείς τιμή όταν κάποιο πρόσωπο κοιτάζει το ρομπότ. Στην συνέχεια οι συναρτήσεις `perfect()`, `decrease()`, `increase()` (Εικόνα 66) καλούν την συνάρτηση `change_face()`, η οποία εκτελεί το κινηματικό μοντέλο και αλλάζει την διάθεση του ρομπότ (Εικόνα 67). Η συνάρτηση `change_state()` (Εικόνα 68) ελέγχει την τιμή του μπροστινού υπερηχητικού αισθητήρα και στην περίπτωση που είναι μικρότερη των 40 εκατοστών οπισθοχωρεί. Στην περίπτωση εντοπισμού προσώπου η συνάρτηση `change_state()` καλεί την συνάρτηση `decrease()` για την αλλαγή της διάθεσης του ρομπότ.

```
def face_detection():
    os.chdir("/home/pi/opencv/data/haarcascades")
    face_cascade = cv2.CascadeClassifier('/home/pi/opencv/data/haarcascades/haarcascade_frontalface_default.xml')
    eye_cascade = cv2.CascadeClassifier('/home/pi/opencv/data/haarcascades/haarcascade_eye.xml')

    cap = cv2.VideoCapture(0)
    flag = 1
    counter = 0
    while flag:
        ret, img = cap.read()
        img = cv2.flip(img, 0)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        if len(faces):
            print(faces)
            counter += 1
            if counter > 4:
                flag = 0
                cap.release()
                return True
```

Εικόνα 65: Συνάρτηση εντοπισμού προσώπου σεναρίου «Ειδική Ενσυναίσθηση»

```

class sen1():
    def __init__(self):
        pass

    def start(self, widget):
        global flag, faces
        faces = 5
        widget.stackedWidget.setCurrentWidget(widget.senariol)
        widget.faces_1.setCurrentWidget(widget.face5_1)
        if (flag):
            self.senl_no.clicked.connect(partial(sen1.perfect, widget))
            self.senl_yes.clicked.connect(partial(sen1.decrease, widget))
            flag = False

    def perfect(widget):
        global faces
        faces = 1
        wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
        play_obj = wave_obj.play()
        play_obj.wait_done()
        senl.change_face(faces, widget)

    def decrease(widget):
        global faces
        if (faces == 1):
            return
        faces = faces - 1
        wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
        play_obj = wave_obj.play()
        play_obj.wait_done()
        senl.change_face(faces, widget)

```

Εικόνα 66: Κλάση σεναρίου «Ειδική Ενσυναίσθηση»

```

def kinematic(mode):
    if mode > 5:
        servo.servo_move(5, 60)
        servo.servo_move(0, 160)           #id, angle
        servo.servo_move(1, 160)
        led.set_led(255,0,0)              #led red color
        wave_obj = sa.WaveObject.from_wave_file("sounds/wrong.wav")
        play_obj = wave_obj.play()
    else:
        servo.servo_move(5, 100)
        servo.servo_move(0, 60)           #id, angle
        servo.servo_move(1, 60)
        led.set_led(0,255,0)              #led green color
        wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
        play_obj = wave_obj.play()

```

Εικόνα 67: Συνάρτηση κινηματικού μοντέλου σεναρίου «Ειδική Ενσυναίσθηση»

```

def increase(widget):
    global faces
    if (faces == 5):
        return
    faces = faces + 1
    wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
    play_obj = wave_obj.play()
    play_obj.wait_done()
    sen1.change_face(faces, widget)

def change_state(widget):
    if ultra.ultra_sonic_front(40):
        stepper.backward(10,10,20)
    if face_detection.face_detection():
        sen1.decrease(widget)

def change_face(id, widget):
    if ( id == 1 ):
        widget.faces_1.setCurrentWidget(widget.face1_1)
        sen1.kinematic(2)
    elif ( id == 2 ):
        widget.faces_1.setCurrentWidget(widget.face2_1)
        sen1.kinematic(4)
    elif ( id == 3 ):
        widget.faces_1.setCurrentWidget(widget.face3_1)
        sen1.kinematic(6)
    elif ( id == 4 ):
        widget.faces_1.setCurrentWidget(widget.face4_1)
        sen1.kinematic(8)
    elif ( id == 5 ):
        widget.faces_1.setCurrentWidget(widget.face5_1)
        sen1.kinematic(10)

```

Εικόνα 68: Συναρτήσεις αλλαγής διάθεσης σεναρίου «Ειδική Ενσυναίσθηση»

4.6.2 Σενάριο «Ειδική Κοινωνική Αλληλεπίδραση»

Το συγκεκριμένο σενάριο εστιάζει στην συμπεριφορά των ατόμων με διάγνωση συνδρόμου Άσπεργκερ. Το σύνδρομο Άσπεργκερ είναι μια ισόβια διαταραχή που επηρεάζει το πως το άτομο αντιλαμβάνεται τον κόσμο, επεξεργάζεται τις πληροφορίες που λαμβάνει και αλληλεπιδρά με άλλα άτομα. Αποτελεί όπως και ο κλασσικός αυτισμός μια «διαταραχή φάσματος» επειδή επηρεάζει τα άτομα με πολλούς διαφορετικούς τρόπους και σε ποικίλο βαθμό. Τα άτομα με Άσπεργκερ έχουν λιγότερα προβλήματα με τη λεκτική επικοινωνία και έχουν συνήθως μέση ή και ανώτερη νοημοσύνη. Ενδέχεται να παρουσιάσουν συγκεκριμένα μαθησιακά προβλήματα όπως δυσλεξία, δυσπραξία ή άλλες καταστάσεις όπως διαταραχή ελλειμματικής προσοχής – υπερκινητικότητα (ΔΕΠΥ).

Σκοπός του σεναρίου είναι η ανάπτυξη επικοινωνιακής και κοινωνικής αλληλεπίδρασης, των μαθητών με διαταραχές στο φάσμα του αυτισμού, με άλλα άτομα. Το ρομπότ λειτουργεί ως μεσολαβητής, ο οποίος διευκολύνει την εκδήλωση συναισθημάτων και συμπεριφορών του χρήστη. Το σενάριο αποτελείται από δύο φάσεις, στην πρώτη ο εκπαιδευτικός διηγείται μια ιστορία στην οποία ο μαθητής με σύνδρομο Άσπεργκερ θα πρέπει να συνεργαστεί στην επιλογή καρτών (**Εικόνα 69**), που περιέχουν ζώα, για την εργασία τους. Κατά την δεύτερη φάση ο εκπαιδευτικός εξηγεί τους κανόνες αλληλεπίδρασης με το ρομπότ. Στην οθόνη του ρομπότ εμφανίζονται από κοινού το πρόσωπο και μία απεικόνιση ενός αυτοκόλλητου (**Εικόνα 70**). Υπάρχουν τρεις απεικονίσεις αυτοκόλλητων, μία γάτα, ένας σκύλος και ένα άλογο. Με την έναρξη του σεναρίου ο χρήστης θα πρέπει να διακρίνει το εικονιζόμενο αυτοκόλλητο και να δείξει στην κάμερα του το σωστό. Σε αυτή την περίπτωση η κατάσταση του προσώπου καλυτερεύει κατά μία, ακούγεται ο αντίστοιχος ήχος και το LED γίνεται πράσινο. Σε αντίθετη περίπτωση το πρόσωπο του αποκτά κατά μια κατάσταση πιο λυπημένη από την τωρινή, ακούγεται ο αντίστοιχος ήχος και το LED γίνεται κόκκινο. Η διαδικασία συνεχίζεται άλλες δύο φορές με αλλαγή του αυτοκόλλητου που εμφανίζεται στην οθόνη και της αναμονής της αντίστοιχης εικόνας από τον χρήστη.



Εικόνα 69: Ενδεικτικές κάρτες ζώων σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»



Εικόνα 70: Αυτοκόλλητα σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»

Η συνάρτηση `find_object()` αναγνωρίζει μέσω της Pi camera τις κάρτες με τα εικονιζόμενα ζώα που έχουν επιλεγθεί (Εικόνα 71). Το ποσοστό ομοιότητας κατά την σύγκριση των καρτών με τα μοντέλα που επιλέχθηκαν (γάτα, σκύλος, άλογο) και διαθέτει το OpenCV έχει ρυθμιστεί στο 70%. Με την ταυτοποίηση ενός μοντέλου, η συνάρτηση, επιστρέφει το όνομα του και στην συνέχεια γίνεται η διαχείριση για την αλλαγή κατάστασης.

```
def find_object():
    global stop_detect
    print("[INFO] starting video stream...")
    vs = VideoStream(usePiCamera=True).start()
    time.sleep(2.0)

    while True and stop_detect == False:
        frame = vs.read()
        frame = imutils.resize(frame, width=400)

        (h, w) = frame.shape[:2]
        blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843, (300, 300), 127.5)
        net.setInput(blob)
        detections = net.forward()
        # loop over the detections
        for i in np.arange(0, detections.shape[2]):
            confidence = detections[0, 0, i, 2]

            if confidence > 0.7: #confidence
                idx = int(detections[0, 0, i, 1])
                if CLASSES[idx] in IGNORE:
                    continue
                if CLASSES[idx] == "cat":
                    vs.stop()
                    print("+cat+")
                    return "cat"

                if CLASSES[idx] == "dog":
                    vs.stop()
                    print("+dog+")
                    return "dog"

                if CLASSES[idx] == "horse":
                    vs.stop()
                    print("+horse+")
                    return "horse"

        vs.stop()
        return "None"

def f_stop_object(status):
    global stop_detect
    stop_detect = not status
```

Εικόνα 71: Κώδικας αναγνώρισης ζώων σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»

Η συνάρτηση `start()` είναι υπεύθυνη για την δημιουργία του γραφικού περιβάλλοντος του σεναρίου (Εικόνα 72). Οι συναρτήσεις `perfect()`, `decrease()`, και `increase()` είναι υπεύθυνες για την ενημέρωση αλλαγής των καταστάσεων. Επίσης, αναλαμβάνουν τους ήχους που ακούγονται κατά την αλλαγή κατάστασης και για την αλλαγή του χρώματος των LED μέσω της συνάρτησής `set_led()` που έχει αναπτυχθεί και περιγράφεται στην παράγραφο 4.2.4.

```
from time import sleep
from threading import Thread
import simpleaudio as sa
import led

flag = True
class sen2():
    def __init__(self, widget):
        super(sen2, self).__init__()
        t = Thread( target = self.start(widget) ).start()
        pass

    def start(self, widget):
        global flag, faces
        faces = 5
        led.set_led(0, 0, 255) #Set leds blue
        widget.stackedWidget.setCurrentWidget(widget.senario2)
        widget.faces_2.setCurrentWidget(widget.faces5_2)
        widget.img_2.setCurrentWidget(widget.img1_2)
        if (flag):
            widget.sen2_no.clicked.connect(partial(sen2.perfect, widget))
            widget.sen2_yes.clicked.connect(partial(sen2.decrease, widget))
            flag = False

    def perfect(widget):
        global faces
        faces = 1
        wave1 = sa.WaveObject.from_wave_file("sounds/correct.wav")
        led.set_led(0, 255, 0) #Set leds green
        play1 = wave1.play()
        play1.wait_done()
        sen2.change_face(faces, widget)

    def decrease(widget):
        global faces
        if (faces == 1):
            return
        faces = faces - 1
        wave2 = sa.WaveObject.from_wave_file("sounds/correct.wav")
        led.set_led(0, 255, 0) #Set leds green
        play2 = wave2.play()
        play2.wait_done()
        sen2.change_face(faces, widget)
```

Εικόνα 72: Κλάση σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»

Η συνάρτηση `change_face()` (Εικόνα 73) είναι υπεύθυνη για τη αλλαγή των καταστάσεων (Εικόνα 62) που εμφανίζονται στην οθόνη του ρομπότ.

```

def change_face(id, widget):
    if ( id == 1 ):
        widget.faces_3.setCurrentWidget(widget.face1_2)
    elif ( id == 2 ):
        widget.faces_3.setCurrentWidget(widget.face2_2)
    elif ( id == 3 ):
        widget.faces_3.setCurrentWidget(widget.face3_2)
    elif ( id == 4 ):
        widget.faces_3.setCurrentWidget(widget.face4_2)
    elif ( id == 5 ):
        widget.faces_3.setCurrentWidget(widget.face5_2)

```

Εικόνα 73: Συνάρτηση αλλαγής κατάστασης σεναρίου «Ειδική Κοινωνική Αλληλεπίδραση»

4.6.3 Σενάριο «Εκφοβισμός»

Σκοπός του συγκεκριμένου σεναρίου είναι να διδάξει στους χρήστες την αναγνώριση και αντιμετώπιση του εκφοβισμού που μπορούν να προκαλέσουν τα παιδιά. Κατά την πρώτη φάση ο εκπαιδευτικός παρουσιάζει διάφορες εικόνες που απεικονίζουν ανθρώπους σε επιθετικές συμπεριφορές. Τα παιδιά συζητάνε τα συναισθήματα που τους προκαλούν αυτές οι εικόνες. Επίσης τονίζουν τα προβλήματα που προκαλούνται στη ζωή των παιδιών οι καταστάσεις εκφοβισμού. Στην δεύτερη φάση ο εκπαιδευτικός με τη βοήθεια του ρομπότ προσπαθεί να παρουσιάσει τις συμπεριφορές εκφοβισμού που πιθανόν θα αντιμετωπίσουν τα παιδιά στην ζωή τους. Μέσα από την οθόνη του ρομπότ τα παιδιά μπορούν να σταματήσουν την ανάρμοστη συμπεριφορά του ρομπότ πατώντας το X ή να ενισχύσουν την θετική συμπεριφορά του πατώντας το √ (Εικόνα 63). Αρχικά το ρομπότ βρίσκεται στην εντονότερα εκφοβιστική κατάσταση κατά την οποία υπάρχει έντονη κίνηση, κόκκινος φωτισμός LED, εκνευρισμένη έκφραση στην οθόνη και ακούγεται η αντίστοιχο ήχοι. Όσο βελτιώνεται η κατάσταση εκφοβισμού εκτελεί ολοένα πιο απαλές κινήσεις και ακούγεται ο αντίστοιχος ήχος, βελτιώνονται οι εκφράσεις του προσώπου και η ένδειξη LED πλησιάζει όλο ένα και περισσότερο το πράσινο. Με το πέρας αυτής της φάσης ο εκπαιδευτικός συγκεντρώνει τα παιδιά και συζητούν τα συμπεράσματα τους τόσο για τα χαρακτηριστικά του εκφοβισμού όσο και για τις δράσεις που μπορούν να κάνουν για να σταματήσουν αυτήν τη κατάσταση.

Η συνάρτηση `start()` είναι υπεύθυνη για την δημιουργία του γραφικού περιβάλλοντος του σεναρίου (**Εικόνα 74**). Οι συναρτήσεις `perfect()`, `decrease()`, και `increase()` είναι υπεύθυνες για την ενημέρωση αλλαγής των καταστάσεων.

```
class sen3():
    def __init__(self):
        pass

    def start(self, widget):      #load Ui
        global flag, faces
        faces = 5
        widget.stackedWidget.setCurrentWidget(widget.scenario3)
        widget.faces_3.setCurrentWidget(widget.face5_3)
        if (flag):
            self.sen3_no.clicked.connect(partial(sen3.perfect, widget))
            self.sen3_yes.clicked.connect(partial(sen3.decrease, widget))
            flag = False

    def perfect(widget):
        global faces
        faces = 1
        wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
        play_obj = wave_obj.play()
        play_obj.wait_done()
        sen3.change_face(faces, widget)

    def decrease(widget):
        global faces
        if (faces == 1):
            return
        faces = faces - 1
        wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
        play_obj = wave_obj.play()
        play_obj.wait_done()
        sen3.change_face(faces, widget)

    def increase(widget):
        global faces
        if (faces == 5):
            return
        faces = faces + 1
        wave_obj = sa.WaveObject.from_wave_file("sounds/wrong.wav")
        play_obj = wave_obj.play()
        play_obj.wait_done()
        sen3.change_face(faces, widget)
```

Εικόνα 74: Κλάση σεναρίου «Εκφοβισμός»

Η συνάρτηση `change_face()` (**Εικόνα 75**) αλλάζει τις καταστάσεις του ρομπότ. Η `set_led()` αλλάζει τα χρώματα της στεφάνης led και η `kinematic()` δημιουργεί το κινηματικό μοντέλο με την χρήση των συναντήσεων `servo.servo_speed()` για την κίνηση των σέρβο κινητήρων και οι συναρτήσεις `stepper.forward()`, `stepper.backward()`,

stepper.right(), stepper_left() για την κίνηση των βηματικών κινητήρων. Οι συναρτήσεις έχουν περιγραφεί στις **παραγράφους 4.2.5** και στην **4.2.6** αντίστοιχα.

```
def change_face(id, widget):
    if ( id == 1 ):
        widget.faces_3.setCurrentWidget(widget.face1_3)
        led.set_led(0,255,0)
        sen3.kinematic(2)
    elif ( id == 2 ):
        widget.faces_3.setCurrentWidget(widget.face2_3)
        led.set_led(127,255,0)
        sen3.kinematic(4)
    elif ( id == 3 ):
        widget.faces_3.setCurrentWidget(widget.face3_3)
        led.set_led(255,255,0)
        sen3.kinematic(6)
    elif ( id == 4 ):
        widget.faces_3.setCurrentWidget(widget.face4_3)
        led.set_led(255,127,0)
        sen3.kinematic(8)
    elif ( id == 5 ):
        widget.faces_3.setCurrentWidget(widget.face5_3)
        led.set_led(255,0,0)
        sen3.kinematic(10)

def kinematic(mode):
    stepper.forward(mode,200,20) #speed, steps, ultrasonic trust threshold
    servo.servo_speed(0, 90, 5, 1.5-mode*0.1) #id, angle, step, speed
    stepper.right(mode,50,1,20)
    servo.servo_speed(2, 30, 5, 1.5-mode*0.1)
    stepper.right(mode,50,0,20)
    servo.servo_speed(2, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(0, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(1, 90, 5, 1.5-mode*0.1)
    stepper.left(mode,50,1,20)
    servo.servo_speed(3, 30, 5, 1.5-mode*0.1)
    stepper.left(mode,50,0,20)
    servo.servo_speed(3, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(1, 0, 5, 1.5-mode*0.1)
    stepper.backward(mode,200,20)
```

Εικόνα 75: Συναρτήσεις αλλαγής κατάστασης και κίνηση σεναρίου «Εκφοβισμός»

Η ταχύτητα κίνησης των σέρβο κινητήρων προκύπτει από την μαθητική έκφραση $1.5-mode*0.1$ οπού mode είναι το επίπεδο έντασης του εκνευρισμού με τιμές 10-2.

4.6.4 Σενάριο «Κοινωνική Συμπεριφορά»

Σκοπό αυτού του σεναρίου αποτελεί η εκμάθηση αναγνώρισης κοινωνικών συμπεριφορών από τα παιδιά. Στην πρώτη φάση ο εκπαιδευτικός παρουσιάζει διάφορες εικόνες που

απεικονίζουν ανθρώπους σε επιθετικές αλλά και κοινωνικά σωστές συμπεριφορές. Συζητάνε με τα παιδιά τι συναισθήματα τους προκαλούν αυτές οι εικόνες, και τονίζουν τα προβλήματα που προκαλούνται στη ζωή των παιδιών οι καταστάσεις εκφοβισμού αλλά και οι κοινωνικά σωστές συμπεριφορές. Στην δεύτερη φάση ο εκπαιδευτικός με τη βοήθεια του ρομπότ προσπαθεί να παρουσιάσει διάφορες κοινωνικές συμπεριφορές. Το ρομπότ αντιδρά ανάλογα με το που το ακουμπάνε τα παιδιά και για πόση ώρα. Έτσι, επηρεάζουν τη συμπεριφορά του ρομπότ και τα συναισθήματά του.

Υπάρχουν τρεις αισθητήρες επαφής πάνω στο ρομπότ, ο ένας βρίσκεται στο κεφάλι και άλλοι δύο στην έξω μεριά των ώμων του. Όσο ο μαθητής ακουμπάει το ρομπότ στο κεφάλι του τότε το πρόσωπο στην οθόνη του ρομπότ καλυτερεύει, ανάβει σε πράσινο χρώμα το LED, ακούγεται ο αντίστοιχος ήχος και μειώνεται η ένταση κίνησης. Ενώ για όση ώρα ο χρήστης ακουμπάει το ρομπότ σε κάποιο ή και στα δύο του χέρια αυτό αποκτά πιο λυπημένο πρόσωπο, ανάβουν τα LED σε κόκκινο χρώμα, ακούγεται ο αντίστοιχος ήχος και αυξάνεται η ένταση κίνησης.

Η συνάρτηση `start()` (Εικόνα 76) είναι υπεύθυνη για τα γραφικά. Η `touch_control()` (Εικόνα 77) ανάλογα με το άγγιγμα εκτελεί τις συναρτήσεις αλλαγής καταστάσεων. Συγκεκριμένα με το άγγιγμα του ρομπότ στους ώμους καλυτερεύει η διάθεση του ρομπότ. Αντίθετα με το άγγιγμα του χρήστη στο κεφάλι χειροτερεύει η διάθεση του ρομπότ.

```
class sen4():
    def __init__(self):
        pass

    def start(self, widget):
        global flag
        global faces
        faces = 5
        widget.stackedWidget.setCurrentWidget(widget.scenario4)
        widget.faces_4.setCurrentWidget(widget.face5_4)
        if (flag):
            self.sen4_no.clicked.connect(partial(sen4.perfect, widget))
            self.sen4_yes.clicked.connect(partial(sen4.decrease, widget))
            flag = False
```

Εικόνα 76: Κλάση σεναρίου «Κοινωνική Συμπεριφορά»

Με την εναλλαγή κατάστασης εκτελείται η συνάρτηση `kinematic()` (Εικόνα 78) που εκτελεί το κινηματικό μοντέλο των κινήσεων του ρομπότ ανάλογα με την κατάσταση στην οποία βρίσκεται.

```
def touch_control():
    end = 0
    start = time.time()    #touch time
    while True:
        if touch.touch(23) or touch.touch(24):
            end = time.time()
            if end - start >= 1:
                sen4.decrease(widget)
                start = time.time()
        elif touch.touch(25):
            end = time.time()
            if end - start >= 1:
                sen4.increase(widget)
                start = time.time()
        else:
            led.set_led(0,0,255)    #Set led blue

def perfect(widget):
    global faces
    faces = 1
    wave_obj = sa.WaveObject.from_wave_file("sounds/correct.wav")
    play_obj = wave_obj.play()
    play_obj.wait_done()
    led.set_led(0,255,0)    #Set led green
    sen4.change_face(faces, widget)
```

Εικόνα 77: Συναρτήσεις σεναρίου «Κοινωνική Συμπεριφορά»

```
def change_face(id, widget):
    if ( id == 1 ):
        widget.faces_4.setCurrentWidget(widget.face1_4)
        sen4.kinematic(2)
    elif ( id == 2 ):
        widget.faces_4.setCurrentWidget(widget.face2_4)
        sen4.kinematic(4)
    elif ( id == 3 ):
        widget.faces_4.setCurrentWidget(widget.face3_4)
        sen4.kinematic(6)
    elif ( id == 4 ):
        widget.faces_4.setCurrentWidget(widget.face4_4)
        sen4.kinematic(8)
    elif ( id == 5 ):
        widget.faces_4.setCurrentWidget(widget.face5_4)
        sen4.kinematic(10)

def kinematic(mode):
    stepper.forward(mode,100,20)    #speed, steps, ultrasonic
    servo.servo_speed(0, 70, 5, 1.5-mode*0.1)    #id, angle, step, speed
    stepper.right(mode,30,1,20)
    servo.servo_speed(2, 40, 5, 1.5-mode*0.1)
    stepper.right(mode,30,0,20)
    servo.servo_speed(2, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(0, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(1, 70, 5, 1.5-mode*0.1)
    stepper.left(mode,50,1,20)
    servo.servo_speed(3, 40, 5, 1.5-mode*0.1)
    stepper.left(mode,50,0,20)
    servo.servo_speed(3, 0, 5, 1.5-mode*0.1)
    servo.servo_speed(1, 0, 5, 1.5-mode*0.1)
    stepper.backward(mode,100,20)
```

Εικόνα 78: Συναρτήσεις αλλαγής διάθεσης και κίνηση σεναρίου «Κοινωνική Συμπεριφορά»

4.6.5 Σενάριο «Επίδειξη»

Σκοπός του συγκεκριμένου σεναρίου είναι η ανάδειξη των δυνατοτήτων και η παρουσίαση του ρομπότ. Διατίθενται δύο βασικές λειτουργίες, η πρώτη είναι η μελέτη και επίδειξη των κινηματικών μοντέλων που προτείνονται για χρήση και η δεύτερη είναι η λειτουργία “φωτογράφισης”. Με την έναρξη του σεναρίου (Εικόνα 80) το led του ρομπότ γίνεται μπλε, με το πάτημα του κουμπιού play εκτελείται η συνάρτηση `play_demo()` (Εικόνα 79). Η συνάρτηση θέτει την εικόνα στο κέντρο της οθόνης στην γκρι έκδοση της, θέτει το led κυανό, ακούγεται ο αντίστοιχος ήχος και εκτελείται το κινηματικό μοντέλο. Κατα το πάτημα του κουμπιού photo εκτελείται η συνάρτηση `photo_demo()` (Εικόνα 81). Η συνάρτηση θέτει το led σε μωβ χρώμα και ανοίγει ένα παράθυρο προεπισκόπησης κάμερα. Στην συνέχεια ακούγεται ο αντίστοιχος ήχος για τρία δευτερόλεπτα και στο τέλος του χρόνου τραβάει μία φωτογραφία και την εμφανίζει στο κεντρικό πλαίσιο της οθόνης πάνω από την εικόνα.

```
def play_demo(self, widget):
    global play_again
    if play_again:
        new_dice = QPixmap('images/camera/robot_logo_gray.png') #Set gray photo
        widget.img.setPixmap(new_dice)
        widget.img.show()

        led.set_led(0,255,255)

        wave_obj = sa.WaveObject.from_wave_file("sounds/w.wav")
        play_obj = wave_obj.play()
        play_again = False

        servo.servo_move(0, 160)           #id, angle
        servo.servo_move(1, 160)
        stepper.forward(10,500,20)         #speed, steps, ultrasonic trust threshold
        servo.servo_move(2, 70)
        servo.servo_move(3, 70)
        servo.servo_move(2, 0)
        servo.servo_move(3, 0)
        servo.servo_move(0, 40)
        servo.servo_move(1, 40)
        stepper.right(10,30,1,20)
        servo.servo_move(4, 140)
        stepper.right(10,30,0,20)
        servo.servo_move(4, 90)
        stepper.left(10,30,1,20)
        servo.servo_move(4, 40)
        stepper.left(10,30,0,20)
        servo.servo_move(4, 90)
        servo.servo_move(5, 110)
        servo.servo_move(5, 70)
        servo.servo_move(5, 110)
        servo.servo_move(5, 70)
        servo.servo_move(5, 90)
        stepper.backward(10,500,20)
```

Εικόνα 79: Κινηματικό μοντέλο σεναρίου «Επίδειξη»

```

class demo():
    def __init__(self):
        pass

    def start(self, widget):
        global flag,play_again
        led.set_led(0,255,0)
        if (flag):
            widget.play.clicked.connect(partial(demo.play_demo, self, widget))
            widget.photo.clicked.connect(partial(demo.photo_demo, self, widget))
            flag = False
        new_dice = QPixmap('images/camera/robot_logo.png')      #Initial photo
        widget.img.setPixmap(new_dice)
        widget.img.show()
        play_again = True

```

Εικόνα 80: Κλάση σεναρίου «Επίδειξη»

```

def photo_demo(self, widget):
    global play_again
    led.set_led(255,0,255)          #led purple color
    try:
        play_again = False
        camera = PiCamera()
        camera.start_preview()

        wave_obj = sa.WaveObject.from_wave_file("sounds/camera.wav")      #camera click sound
        play_obj = wave_obj.play()

        end = 0
        start = time.time()
        while True:
            end = time.time()
            if end - start <= 3:      #wait 3 seconds
                led.rainbow_cycle(0.001)    #led rainbow mode
            else:
                camera.capture('images/camera/pro.jpg',resize=(560, 310)) #Camera capture
                camera.stop_preview()
                camera.close()
                break
    except:
        led.set_led(255,0,0)          #led red color
        print("Demo: Error can not open camera")

    led.set_led(255,0,255)
    new_dice = QPixmap('images/camera/pro.jpg')
    widget.img.setPixmap(new_dice)
    widget.img.show()

```

Εικόνα 81: Συνάρτηση λήψης φωτογραφιών σεναρίου «Επίδειξη»

4.7 Σύνοψη

Στο κεφάλαιο αυτό αναλύθηκαν οι δυνατότητες του ρομποτικού συστήματος που εγκαταστάθηκε στον μικροεπεξεργαστή μας. Περιεγράφηκαν οι συναρτήσεις λειτουργίας του ρομπότ. Έγινε αναφορά στην δυνατότητα διασύνδεσης του ρομπότ με το πληροφοριακό σύστημα και πως θα πρέπει να επιτευχθεί η μεταξύ τους επικοινωνία. Τέλος, παρουσιάστηκε ο σκοπός και η υλοποίηση των προ εγκατεστημένων σεναρίων.

Κεφάλαιο 5

Μετά την ανάλυση του ρομποτικού συστήματος που πραγματοποιήθηκε στο προηγούμενο κεφάλαιο, έχει ολοκληρωθεί και η παρουσίαση του ρομπότ.

Στο κεφάλαιο αυτό, γίνεται μια ανακεφαλαίωση με τα γενικά χαρακτηριστικά του συστήματος. Επίσης, παρουσιάζονται οι μετρήσεις, οι οποίες είναι σχετικές με το ρομπότ. Τέλος, εξάγονται τα συμπεράσματα που προέκυψαν μετά την υλοποίηση του ρομπότ αλλά και οι μελλοντικές επεκτάσεις που θα μπορούσαν να υπάρξουν.

5.1 Σύνοψη

Η έλλειψη ενσυναίσθησης των ατόμων οδηγεί σε ανάρμοστες συμπεριφορές προς τα άτομα με διαταραχές του αυτιστικού φάσματος. Έτσι, γεννήθηκε η ιδέα για την κατασκευή ενός ρομποτικού συστήματος, το οποίο θα χρησιμοποιείται σαν εργαλείο από εκπαιδευτές, που θα απευθύνεται σε παιδιά ηλικίας τεσσάρων με εννέα ετών με σκοπό την καλλιέργεια της ενσυναίσθησης.

Το ρομπότ έχει κατασκευαστεί από πλαστικά εκτυπωμένα σε τρισδιάστατο εκτυπωτή. Βασική απαίτηση ήταν να είναι ασφαλές και φιλικό προς τα παιδιά, με λείες ακμές για να αποφευχθεί κάποιος τραυματισμός.

Ο υλικός εξοπλισμός του ρομπότ μελετήθηκε με τέτοιο τρόπο ώστε να καλύπτονται όλες οι απαιτήσεις του ρομποτικού συστήματος. Το ρομπότ διαθέτει αισθητήρια όργανα ώστε να αντιλαμβάνεται τις μεταβολές στο περιβάλλον του. Το λογισμικό που αναπτύχθηκε αξιοποιεί πλήρως τα αισθητήρια του ρομπότ και είναι σε θέση να διατηρεί την ασφάλεια των χρηστών του.

Σε συνεργασία με το Παιδαγωγικό Τμήμα Νηπιαγωγών του Πανεπιστημίου Δυτικής Μακεδονίας δημιουργήθηκαν σενάρια που βοηθούν τα παιδιά να κατανοούν και να ερμηνεύουν τα συναισθήματα των ανθρώπων. Υλοποιήθηκαν πέντε σενάρια (Ειδική Ενσυναίσθηση, Ειδική Κοινωνική Αλληλεπίδραση, Εκφοβισμός, Κοινωνική Συμπεριφορά, Επίδειξη), τα οποία βρίσκονται προ εγκατεστημένα στο ρομπότ. Επίσης, το ρομπότ έχει την δυνατότητα διασύνδεσής με πληροφοριακό σύστημα από το οποίο οι χρήστες θα έχουν την δυνατότητα να προγραμματίσουν και να προσθέσουν τα δικά τους σενάρια αξιοποιώντας πλήρως τους κινητήρες και τα αισθητήρια όργανα του ρομπότ.

5.2 Μετρικά Συστήματος

Το ρομπότ αποτελείται, από το υλικό μέρος, όσο και από λογισμικό που αναπτύχθηκε. Παρατίθενται κάποιες μετρήσεις που αφορούν τόσο την τελική κατασκευή όσο και τον κώδικα.

Για την κατασκευή του ρομπότ εξήχθησαν από το πρόγραμμα σχεδίασης 35 αρχεία τύπου STL (Standard Triangle Language). Για την εκτύπωση χρειάστηκαν 16 αρχεία τύπου gcode (NC - γλώσσα προγραμματισμού αριθμητικού ελέγχου). Ο συνολικός χρόνος εκτύπωσης που χρειάστηκε ήταν 142 ώρες και περίπου 3,5 κιλά νήματος πλαστικού (filament). Χρειάστηκαν περίπου 2 ώρες για την συναρμολόγηση του ρομπότ.

Το λογισμικό του ρομπότ αποτελείται από 25 αρχεία τύπου python (1479 γραμμές κώδικα) τα οποία ενσωματώνουν τα γράφηκα της εφαρμογής (3286 γραμμές κώδικα). Συγκεκριμένα, 169 KB είναι τα αρχεία που αναπτύχθηκαν και 315 KB οι φωτογραφίες που χρησιμοποιήθηκαν. Τέλος ο χρόνος που χρειάζεται για να εκτελέσει ο μικροεπεξεργαστής την εφαρμογής ανέρχεται στα 1,5 δευτερόλεπτα.

5.3 Ανάλυση SWOT

Στην ενότητα αυτή εξετάζονται τα Δυνατά Σημεία (Strengths), οι Αδυναμίες (Weaknesses), οι Ευκαιρίες (Opportunities) και οι Απειλές (Threats) του ρομποτικού συστήματος που αναπτύχθηκε.

5.3.1 Δυνατά Σημεία

Ένα από τα δυνατά σημεία του ρομπότ είναι ο τρόπος κατασκευής του. Τα πλαστικά του ρομπότ είναι εκτυπωμένα σε τρισδιάστατο εκτυπωτή. Έτσι, δίνεται η δυνατότητα εκτύπωσης επιπλέον εξαρτημάτων, με σκοπό την επέκταση των δυνατοτήτων του, καθώς και η τροποποίηση των ήδη υπάρχοντων.

Ο ελεγκτής που χρησιμοποιήθηκε δίνει την δυνατότητα να εξελιχθεί σε μεγάλο βαθμό το ρομποτικό σύστημα. Το Raspberry Pi εντάσσεται στο πλαίσιο του ανοιχτού λογισμικού, που σημαίνει ότι η κοινότητα αναβαθμίζει διαρκώς τις δυνατότητες του. Χαρακτηριστικό που ξεχωρίζει το ρομπότ μας από τα υπόλοιπα ρομπότ της κατηγορίας του, είναι η οθόνη αφής που διαθέτει στο κεφάλι του. Μέσω της οθόνης γίνεται ο έλεγχος του ρομπότ, η εκτέλεση των προ

εγκατεστημένων σεναρίων που διαθέτει και η εκτέλεση των παιχνιδιών που παρέχονται μέσω του εργαλείου gcompris.

Επίσης, δίνεται η δυνατότητα διασύνδεση με πληροφοριακό σύστημα και εφαρμογή τηλεχειρισμού που επεκτείνουν τις λειτουργίες και τις χρήσεις του ρομπότ.

5.3.2 Αδυναμίες

Η βασική αδυναμία του συστήματος είναι ότι το ρομποτικό σύστημα για να είναι σε θέση να ενταχθεί σε κάποια εκπαιδευτική μονάδα, θα πρέπει να γίνουν αρκετές δοκιμές προκειμένου να τεκμηριωθεί η ασφάλεια των χρηστών του. Εξίσου σημαντική είναι η επιβεβαίωση, σε συνεργασία με ειδικούς, της επιτυχούς προσομοίωσης του κώδικα που αναπτύχθηκε ως προς τα προ εγκατεστημένα σενάρια και τις απαιτήσεις που θα πρέπει να πληρούν.

Για την κατασκευή απαιτείται αρκετός χρόνος εκτύπωσης. Θα μπορούσαν το κομμάτια να σχεδιαστούν με τέτοιο τρόπο ώστε να μειωθεί τόσο ο απαιτούμενος χρόνος εκτύπωσης αλλά και το υλικό (filament) που χρειάζεται.

5.3.3 Ευκαιρίες

Εξαιτίας των γρήγορων ρυθμών της σύγχρονης εποχής έχει παρατηρηθεί έλλειψη της ενσυναίσθησης. Η παρούσα εργασία και η κατασκευή του ρομποτικού συστήματος δίνει την δυνατότητα αντιμετώπισης του προβλήματος. Η καλλιέργεια της ενσυναίσθησης θα δώσει την δυνατότητα αποφυγής ανάρμοστων συμπεριφορών σε άτομα που αντιμετωπίζουν νευρο-αναπτυξιακές διαταραχές. Τέλος θα βοηθήσει στην ανάπτυξη και εξέλιξη των παιδιών δεδομένου ότι θα είναι σε θέση να αντιλαμβάνονται και να έχουν ως οδηγό σκέψης τις αντιδράσεις και τα συναισθήματα των ατόμων που συναναστρέφονται.

5.3.4 Απειλές

Σημαντικό ρολό στην εξέλιξη ρομποτικών συστημάτων με σκοπό την καλλιέργεια της ενσυναίσθησης παίζει η κοινωνία, η οποία θα πρέπει να κατανοήσει και να αποδεχτεί τα οφέλη και να βοηθήσει στην ένταξη αυτών των συστημάτων στις εκπαιδευτικές μονάδες.

Η απειλή από εταιρίες, που δραστηριοποιούνται χρόνια στην ανάπτυξη εργαλείων με σκοπό την καλλιέργεια της ενσυναίσθησης, παίζουν καθοριστικό ρόλο στην περεταίρω ανάπτυξη του ρομπότ και των εργαλείων που διαθέτει.

5.4 Ασφάλεια

Η ασφάλεια του ρομποτικού συστήματος έπαιξε καθοριστικό ρολό στην μελέτη και στον σχεδιασμό του. Η επιλογή όλων των υλικών εξαρτημάτων του ρομπότ έγινε με γνώμονα την ασφάλεια των χρηστών που έρχονται σε επαφή με αυτό αλλά και του χώρου που βρίσκεται.

Ο σχεδιασμός και η επιλογή λείων επιφανειών εξαλείφει τυχόν τραυματισμούς που μπορεί να προκληθούν. Από πλευράς λογισμικού έχουν περιοριστεί οι ταχύτητες όλων των κινητήρων. Έχουν προστεθεί αισθητήρια όργανα, με τα οποία το ρομπότ αντιλαμβάνεται τις αλλαγές του περιβάλλοντος και παράλληλα διασφαλίζεται η ασφάλεια των χρηστών αλλά και του ίδιου του μηχανήματος.

5.5 Μελλοντικές Επεκτάσεις

Παρακάτω αναφέρονται μερικές βελτιώσεις και στην συνέχεια μελλοντικές επεκτάσεις, που αφορούν την παρούσα διπλωματική εργασία.

Με την συμβολή ειδικά καταρτισμένων ατόμων στον τομέα της παιδικής ψυχολογίας θα μπορούσε να βελτιωθεί το γραφικό περιβάλλον χρήστη (GUI), τα χρώματα, τα σχήματα καθώς και οι εικόνες που χρησιμοποιούνται εντός του λογισμικού, προκειμένου να γίνουν ακόμα πιο θελκτικά προς τα παιδιά. Επίσης, θα μπορούσε να σχεδιαστεί με τέτοιο τρόπο ώστε να έχει περισσότερες καμπύλες.

Στην παρούσα φάση η σύνδεση των περιφερειακών εξαρτημάτων του ρομποτικού συστήματος γίνεται με την βοήθεια ενός breadboard, το οποίο αποτελεί μια γρήγορη λύση σύνδεσης. Σαν μελλοντική επέκταση θα μπορούσε να μελετηθεί και να υλοποιηθεί η δημιουργία μιας πλακέτας τυπωμένου κυκλώματος με βασικό πλεονέκτημα την μείωση των καλωδιώσεων και των επιμέρους διαφορετικών πλακετών.

Υπάρχει η δυνατότητα προσθήκης περισσότερων αισθητήρων και εξαρτημάτων για τον εμπλουτισμό των δυνατοτήτων του ρομπότ. Θα μπορούσαν να σχεδιαστούν εξαρτήματα, τα οποία θα εκτυπωθούν σε τρισδιάστατο εκτυπωτή και θα ενσωματωθούν στο ρομπότ.

Το σύστημα μας έχει σχεδιαστεί να υποστηρίζει την επικοινωνία με ένα πληροφοριακό σύστημα, στο οποίο ο χρήστης μπορεί να παρακολουθήσει την κατάσταση του ρομποτικού συστήματος. Επίσης μέσω του συστήματος, ο χρήστης θα έχει την δυνατότητα δημιουργίας και εκτέλεσης των δικών του σεναρίων στο ρομπότ.

Τέλος το ρομποτικό σύστημα διαθέτει τις απαραίτητες συναρτήσεις κίνησης προκειμένου να μπορεί να επικοινωνήσει με κάποια εφαρμογή τηλεχειρισμού. Έτσι υπάρχει η δυνατότητα δημιουργίας μιας εφαρμογής τηλεχειρισμού για κινητές συσκευές. Η συγκεκριμένη εφαρμογή θα μπορούσε να επικοινωνεί είτε μέσω τοπικού δικτύου είτε μέσω του προτύπου επικοινωνίας bluetooth, καθώς το Raspberry Pi 4 διαθέτει αντίστοιχο πρότυπο επικοινωνίας.

5.6 Προβλήματα που αντιμετωπίστηκαν

Η συγκεκριμένη εργασία λόγω της δομής της, αποτελούμενη από υλικό και λογισμικό μέρος, δημιούργησε χώρο για την εμφάνιση προβλημάτων διαφόρων καταστάσεων. Τα κυριότερα από αυτά αφορούσαν την σωστή λειτουργία των εξαρτημάτων τα οποία θα έπρεπε να λειτουργούν ομαλά προκειμένου να ολοκληρωθούν όλες οι απαιτήσεις που είχαν τεθεί.

Αρχικά ο επιλεγμένος οδηγός βηματικών κινητήρων ήταν ο A4988, ο οποίος επιλέχθηκε λόγω του χαμηλού κόστους και του μικρού χώρου που καταλαμβάνει. Κατά τις δοκιμές λειτουργίας του σε συνδυασμό με του βηματικούς κινητήρες που χρησιμοποιήσαμε διαπιστώθηκε η αστάθεια και ανωμαλία αλλαγής των φάσεων των βηματικών κινητήρων. Αυτό είχε ως συνέπεια την δόνηση της κατασκευής κατά την λειτουργία του. Αυτή η κατάσταση όμως ήταν ανεπιθύμητη καθώς προκαλούσε δυνατό θόρυβο με τον συντονισμό των πλαστικών κομματιών του ρομπότ και αμέσως έκανε αποτρεπτική την χρήση από παιδιά ηλικίας τεσσάρων με εννέα ετών. Υπήρξαν αρκετές προσπάθειες βελτιστοποίησης του προβλήματος με την παρέμβαση στον κώδικα. Παρατηρήθηκε μικρή βελτίωση της κατάστασης με την χρήση πλακέτα εξομάλυνσης (smoother) της κίνηση. Παρόλα αυτά δεν επιτεύχθηκε το επιθυμητό αποτέλεσμα, έτσι αποφασίσαμε να αλλάξουμε τους οδηγούς των κινητήρων. Οι TB6560 είναι οι οδηγοί βηματικών κινητήρων που τελικά χρησιμοποιήθηκαν.

Κατά την συναρμολόγηση των επιμέρους περιφερειακών στοιχείων του ρομπότ με τα τρισδιάστατα εκτυπωμένα τμήματα διαπιστώθηκε ότι η λειτουργία των χωρητικών αισθητήρων αφής επηρεάζεται σημαντικά από την πυκνότητα και πάχος του υλικού. Το πρόβλημα λύθηκε με τον κατάλληλο σχεδιασμό του υλικού σύμφωνα με τις απαιτήσεις λειτουργίας των αισθητήρων.

Στην αρχική μελέτη για τα εξαρτήματα καταγραφής ήχου επιλέχθηκε ένα μικρόφωνο χωρίς κύκλωμα ενίσχυσης. Διαπιστώθηκε κατά την δοκιμή ότι υπήρχε θέμα στην ευαισθησία του

συγκεκριμένου μικροφώνου. Έτσι αντικαταστάθηκε το μικρόφωνο με νέο, το οποίο διαθέτει κύκλωμα ενίσχυσης ήχου.

Στην φάση των δοκιμών του συστήματος διαπιστώθηκε υψηλή θερμοκρασία στον μικροεπεξεργαστή, για τον λόγο αυτό σχεδιάστηκαν αεραγωγοί στην περιοχή τοποθέτησης του μικροεπεξεργαστή.

5.7 Στάδια υλοποίησης

Ύστερα από έρευνα που πραγματοποιήθηκε για την ανάπτυξη του ρομποτικού συστήματος, κρίθηκε αναγκαίος ο καταμερισμός των σταδίων υλοποίησης.

Πρώτη ομάδα σταδίων:

- Αναζήτηση απαιτούμενων λογισμικών και εργαλείων.
- Ανάλυση των περιπτώσεων χρήσης.
- Απαιτήσεις σχεδιασμού εμφάνιση του ρομπότ.
- Ανάπτυξη γραφικού περιβάλλοντος.
- Δημιουργία εκτελέσιμου αρχείου της εφαρμογής.
- Ανάπτυξη σεναρίων «Ειδική ενσυναίσθηση» και «Ειδική κοινωνική αλληλεπίδραση».

Δεύτερη ομάδα σταδίων:

- Έρευνά για τον απαραίτητο εξοπλισμό.
- Έρευνά για τις ενεργειακές απαιτήσεις του ρομποτικού συστήματος.
- Σύνδεση και δοκιμή της ομαλής λειτουργίας του εξοπλισμού.
- Προετοιμασία μικροεπεξεργαστή.
- Δημιουργία αρχείων λειτουργία του εξοπλισμού.
- Δημιουργία υποδομής για διασύνδεση με πληροφοριακό σύστημα.
- Ανάπτυξη σεναρίων «Έκφοβισμός», «Κοινωνική συμπεριφορά» και «Επίδειξη».

Βιβλιογραφία

- [1] **ΓΕΩΡΓΙΑ ΓΕΩΡΓΙΤΣΑΝΑΚΟΥ, ΓΑΡΥΦΑΛΙΑ ΧΑΡΙΤΑΚΗ** Ανάπτυξη της Ενσυναίσθησης σε παιδιά προσχολικής ηλικίας τυπικής ανάπτυξης [Συνέδριο] : Πανελλήνιο Συνέδριο Επιστημών Εκπαίδευσης, 2017. - σσ. 117-136.
- [2] **Michael Cole, Sheila Cole, Cynthia Lightfoot** Ανάπτυξη των Παιδιών [Βιβλίο] / μεταφρ. Κουλεντιανού Μαργαρίτα. : Gutenberg, 2015.
- [3] **Embodied Moxie** [Ηλεκτρονικό] - 7 2020. - <https://embodied.com/products/moxie-reservation>
- [4] **Luxai** [Ηλεκτρονικό] - 7 2020. - <https://luxai.com/>
- [5] **Luxai QTrobot** [Ηλεκτρονικό] - 7 2020. - <https://luxai.com/qtrobot-for-research/>
- [6] **Python** [Ηλεκτρονικό] - 8 2020 - <https://www.python.org/doc/essays/blurb/>
- [7] **Python** [Ηλεκτρονικό] - 8 2020 - <https://www.python.org/about/>
- [8] **OpenCV** [Ηλεκτρονικό] - 8 2020 - https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html
- [9] **Python** [Ηλεκτρονικό] - 8 2020 - <https://docs.python.org/3/glossary.html#term-virtual-environment>
- [10] **Riverbank Computing** [Ηλεκτρονικό] - 8 2020 - <https://riverbankcomputing.com/software/pyqt/intro>
- [11] **Qt** [Ηλεκτρονικό] - 8 2020 - https://wiki.qt.io/About_Qt
- [12] **GCompris** [Ηλεκτρονικό] - 8 2020 - <https://gcompris.net/index-el.html>
- [13] **Mobaxterm** [Ηλεκτρονικό] - 8 2020 - <https://mobaxterm.mobatek.net/>
- [14] **Nano** [Ηλεκτρονικό] - 8 2020 - <https://www.nano-editor.org/dist/latest/nano.1.html>
- [15] **Notepad++** [Ηλεκτρονικό] - 8 2020 - <https://notepad-plus-plus.org/>
- [16] **Visual Studio** [Ηλεκτρονικό] - 8 2020 - <https://code.visualstudio.com/docs/editor/whyvscode>
- [17] **Autodesk Fusion 360** [Ηλεκτρονικό] - 8 2020 - <https://www.autodesk.com/products/fusion-360/overview>
- [18] **Prusa** [Ηλεκτρονικό]. - 8 2020 - <https://www.prusa3d.com/original-prusa-i3-mk3/>
- [19] **Raspberry Pi** [Ηλεκτρονικό] - 8 2020 - <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>

- [20] **Raspberry Pi** Touch Display [Ηλεκτρονικό] - 8 2020 - <https://www.raspberrypi.org/documentation/hardware/display/>
- [21] **Raspberry Pi** Camera Module V2 [Ηλεκτρονικό] - 8 2020 - <https://www.raspberrypi.org/blog/new-8-megapixel-camera-board-sale-25/>
- [22] **Texas Instruments** LM2596 [Ηλεκτρονικό] - 8 2020 - <https://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [23] **D Young Hugh** Πανεπιστημιακή φυσική [Βιβλίο] : Εκδόσεις Παπαζήση, 1995. - σ. 1350.
- [24] **Components101** NEMA17 [Ηλεκτρονικό] - 8 2020 - https://components101.com/sites/default/files/component_datasheet/NEMA17.pdf
- [25] **Allelectronics** SMC-5 [Ηλεκτρονικό] - 8 2020 - https://www.allelectronics.com/mas_assets/media/allelectronics2018/spec/SMC-5.pdf
- [26] **Electronicoscaldas** MG995 Tower Pro [Ηλεκτρονικό] - 8 2020 - https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf
- [27] **Adafruit** PCA9685 [Ηλεκτρονικό] - 8 2020 - <https://cdn-learn.adafruit.com/downloads/pdf/16-channel-pwm-servo-driver.pdf>
- [28] **J Chapman Stephen** Ηλεκτρικές μηχανές AC-DC [Βιβλίο] / επιμ. Θεοδοουλίδης Θεόδωρος Π. : Εκδόσεις Τζιόλα, 2019.
- [29] **Elecrow** Capacitive Touch Sensor [Ηλεκτρονικό] - 8 2020 - <https://www.electroschematics.com/capacitive-touch-sensor/>
- [30] **Figliola, Beasley** ΤΕΧΝΙΚΕΣ ΜΕΤΡΗΣΕΙΣ [Βιβλίο] : ΓΡΗΓΟΡΙΟΣ ΧΡΥΣΟΣΤΟΜΟΥ ΦΟΥΝΤΑΣ, 2014.
- [31] **Sparkfun** HCSR04 [Ηλεκτρονικό] - 8 2020 - <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [32] **Vishay** Tcrt5000 [Ηλεκτρονικό] - 8 2020 - <https://www.vishay.com/docs/83760/tcrt5000.pdf>
- [33] **Adafruit** WS2812 [Ηλεκτρονικό] - 8 2020 - <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [34] **Corp Diamond Systems** COM-Based SBCs: The Superior Architecture for Small Form Factor Embedded Systems [Ηλεκτρονικό] - 8 2020 - <http://whitepaper.opsy.st/WhitePaper.diamondsys-combased-sbcs-wpfinal-.pdf>

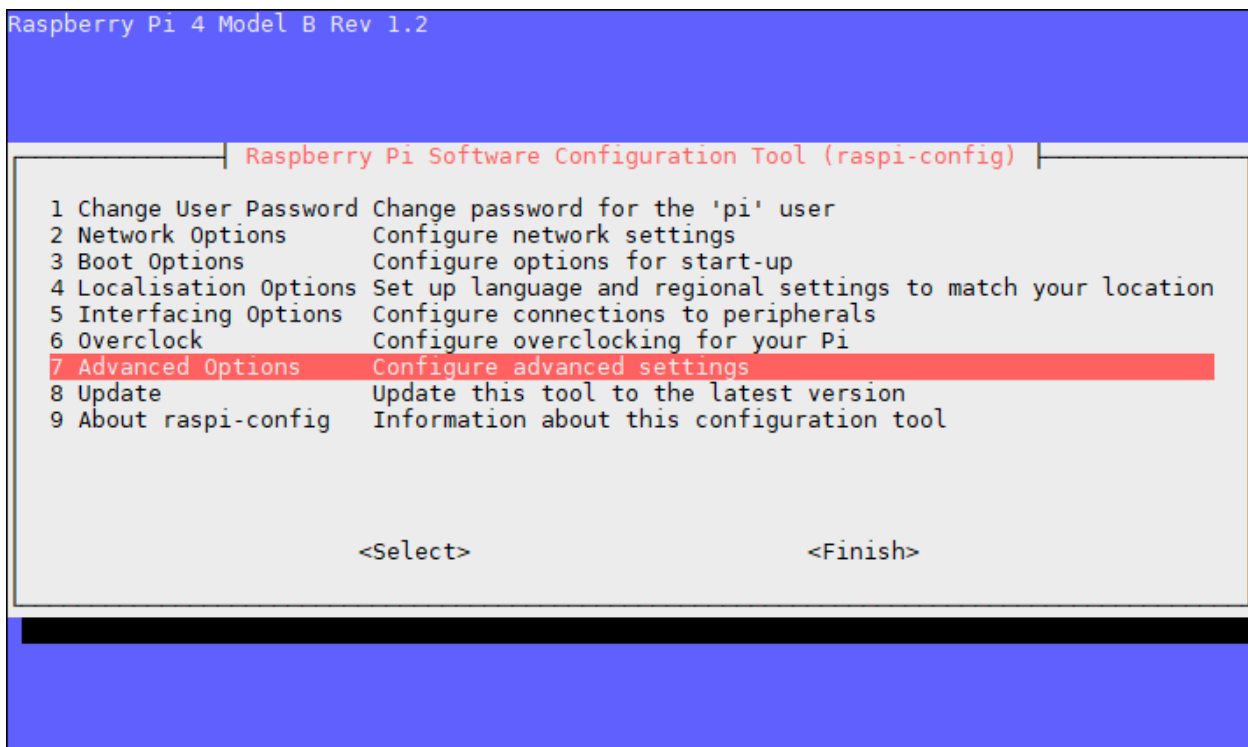
Παράρτημα Α – Εγκατάσταση OpenCV

Στο παράρτημα αυτό παρουσιάζεται ο τρόπος εγκατάστασης του εργαλείου OpenCV στο Raspberry Pi. Αρχικά θα πρέπει να υπάρχει προεγκατεστημένο το λειτουργικό σύστημα Raspbian Buster σε μια κάρτα μνήμης τουλάχιστον 16GB. Έπειτα προετοιμάζουμε το σύστημα με τις παρακάτω ενέργειες.

Βήμα 1: Ανάπτυξη του συστήματος αρχείων και ανάκτηση χώρου

Εκτελούμε την εντολή `sudo raspi-config` μέσω της οποίας θα επεκτείνουμε το σύστημα αρχείων.

Επιλέγουμε την έβδομη επιλογή «Advanced Options» (**Εικόνα 82**) και στην συνέχεια επιλέγουμε την πρώτη εντολή «Expand Filesystem».



```
Raspberry Pi 4 Model B Rev 1.2
Raspberry Pi Software Configuration Tool (raspi-config)
1 Change User Password Change password for the 'pi' user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

<Select>                <Finish>
```

Εικόνα 82: Οθόνη διαμόρφωσης του Raspbian Buster (raspi-config)

Έπειτα πατάμε το δεξί βελάκι από το πληκτρολόγιο και επιλέγουμε το κουμπί «Finish». Ακόλουθα εκτελούμε την εντολή `sudo reboot` για την επανεκκίνηση του συστήματος. Αν και δεν απαιτείται, προτείνεται η διαγραφή τόσο του Wolfram Engine όσο και του LibreOffice για ανάκτηση περίπου 1 GB χώρου στο Raspberry Pi με την εκτέλεση των ακόλουθων εντολών:

```
sudo apt-get purge wolfram-engine
sudo apt-get purge libreoffice *
sudo apt-get clean
sudo apt-get autoremove
```

Βήμα 2: Εγκατάσταση εξαρτήσεων

Οι ακόλουθες εντολές θα ενημερώσουν και θα αναβαθμίσουν τυχόν υπάρχοντα πακέτα, ακολουθούμενα από την εγκατάσταση εξαρτήσεων, βιβλιοθηκών εισόδου/εξόδου και πακέτων βελτιστοποίησης για το OpenCV:

Το πρώτο βήμα είναι η ενημέρωση και να αναβάθμιση τυχόν υπαρχόντων πακέτων με την εκτέλεση των εντολών:

```
sudo apt-get update && sudo apt-get upgrade
```

Στη συνέχεια, πρέπει να εγκαταστήσουμε ορισμένα εργαλεία προγραμματιστή, συμπεριλαμβανομένου του CMake, το οποίο μας βοηθά να διαμορφώσουμε τη διαδικασία δημιουργίας OpenCV μέσω της εντολής:

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

Στη συνέχεια, πρέπει να εγκαταστήσουμε μερικά πακέτα εισόδου/εξόδου εικόνας που μας επιτρέπουν να φορτώνουμε διάφορες μορφές αρχείων εικόνας από το δίσκο. Παραδείγματα τέτοιων μορφών αρχείων περιλαμβάνουν JPEG, PNG, TIFF, κ.λπ. .:

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev
```

Χρειαζόμαστε επίσης πακέτα εισόδου/εξόδου βίντεο. Αυτές οι βιβλιοθήκες μας επιτρέπουν να διαβάζουμε διάφορες μορφές αρχείων βίντεο από το δίσκο, καθώς και να δουλεύουμε απευθείας με ροές βίντεο:

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
sudo apt-get install libxvidcore-dev libx264-dev
```

Η βιβλιοθήκη OpenCV συνοδεύεται από μια υπό-ενότητα που ονομάζεται `highgui` χρησιμοποιείται για την εμφάνιση εικόνων στην οθόνη μας και τη δημιουργία βασικών GUI. Για

να μεταγλωττίσουμε την ενότητα `highgui`, πρέπει να εγκαταστήσουμε τη βιβλιοθήκη ανάπτυξης GTK και τις προϋποθέσεις:

```
sudo apt-get install libfontconfig1-dev libcairo2-dev
sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev
sudo apt-get install libgtk2.0-dev libgtk-3-dev
```

Πολλές λειτουργίες εντός του OpenCV (συγκεκριμένα `matrix`) μπορούν να βελτιστοποιηθούν περαιτέρω εγκαθιστώντας μερικές επιπλέον εξαρτήσεις, οι οποίες είναι ιδιαίτερα σημαντικές για συσκευές με περιορισμούς πόρων, όπως το Raspberry Pi:

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

Οι ακόλουθες προϋποθέσεις αφορούν το Βήμα 4α. Πρόκειται για σύνολα δεδομένων HDF5 και Qt GUI:

```
sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5- 103
sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
```

Τέλος εγκαθιστούμε κάποιες διασυνδέσεις της γλώσσας προγραμματισμού Python 3:

```
sudo apt-get install python3-dev
```

Εγκαθιστούμε το εργαλείο `pip` χρησιμοποιώντας τις ακόλουθες εντολές:

```
sudo apt-get install python3-pip
```

Εγκαθιστούμε την βιβλιοθήκη `imutils` η οποία παρέχει τις βασικές λειτουργίες επεξεργασίας εικόνας όπως περιστροφή, αλλαγή μεγέθους, εμφάνιση εικόνων `Matplotlib`, ταξινόμηση περιγραμμάτων, ανίχνευση άκρων με Python 3.

```
sudo pip3 install imutils
```

Βήμα 3: Δημιουργία του εικονικού περιβάλλοντος Python και εγκατάσταση του NumPy

Εγκαθιστούμε το `virtualenv` και το `virtualenvwrapper`:

```
sudo pip install virtualenv virtualenvwrapper
```

Ανοίγουμε προς επεξεργασία το αρχείο `~/.bashrc` με τον κειμενογράφο `nano`

`nano ~/.bashrc` και προσαρτήσουμε την κάτω μεριά του αρχείου τις εντολές:

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
export VIRTUALENVWRAPPER_VIRTUALENV=/usr/local/bin/virtualenv
```

```
source /usr/local/bin/virtualenvwrapper.sh
export VIRTUALENVWRAPPER_ENV_BIN_DIR=bin
```

Αποθηκεύουμε με τον συνδυασμό Ctrl+X εν συνεχεία πατάμε Y και Enter. Φορτώνουμε ξανά το αρχείο με την εντολή `source ~/.bashrc` για να εφαρμοστούν οι αλλαγές στην τρέχουσα συνεδρία bash.

Δημιουργούμε το εικονικό περιβάλλον Python 3:

```
mkvirtualenv cv -p python3
```

Τέλος, λόγω της χρήσης μια μονάδα Raspberry Pi Camera η οποία είναι συνδεδεμένη στο RPi, θα πρέπει επίσης να εγκατασταθεί το API PiCamera :

```
pip install "picamera [array]"
```

Βήμα 4: Εγκατάσταση OpenCV 4 με την εντολή pip

Με την παρακάτω εντολή εγκαθίσταται το OpenCV τέταρτης έκδοσης

```
pip install opencv-contrib-python == 4.1.0.25
pip install opencv-python
```

Βήμα 5: Δοκιμή της εγκατάστασης OpenCV 4 Raspberry Pi BusterOS

Προκειμένου να γίνει ένας γρήγορος έλεγχος σωστής εγκατάστασης, αποκτάμε πρόσβαση στο εικονικό περιβάλλον cv, ενεργοποιούμε ένα κέλυφος Python και εισάγουμε τη βιβλιοθήκη OpenCV με τις παρακάτω εντολές:

```
$ cd ~
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'4.1.1'
>>>
```

Παράρτημα Β – Εγκατάσταση βιβλιοθηκών ήχου, του PyQt και του gcompris

Ελέγχουμε και ενημερώνουμε τις βιβλιοθήκες του συστήματος μας εκτελώντας τις εντολές:

```
sudo apt-get update && sudo apt-get upgrade
```

Εγκαθιστούμε τις βιβλιοθήκες και τα εργαλεία του προγράμματος PyQt5

```
sudo apt-get install qt5-default
```

```
sudo apt-get install pyqt5-dev-tools
```

Εγκαθιστούμε την βιβλιοθήκη pulseaudio η οποία προορίζεται για την διαχείριση εντολών ήχου καθώς και τις απαιτούμενες βιβλιοθήκες και εργαλεία για την σωστή λειτουργία του ήχου στο Raspberry Pi.

```
sudo apt-get install pulseaudio
```

```
sudo pip3 install simpleaudio
```

```
sudo apt-get install libgstreamer1.0-0 gstreamer1.0-plugins-base gstreamer1.0-  
plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-  
libav gstreamer1.0-doc gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa  
gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-pulseaudio
```

Για την εγκατάσταση της εφαρμογής παιχνιδιών εκτελούμε την παρακάτω εντολή και επανεκκινούμε το σύστημα.

```
sudo apt-get install gcompris
```

```
sudo reboot
```


Παράρτημα Γ – Εγκατάσταση βιβλιοθηκών για LED, κινητήρες και WiFi

Για την εγκατάσταση των βιβλιοθηκών των LED και της πλακέτας των σέρβο κινητήρων εκτελούμε τα παρακάτω

```
sudo pip3 install rpi_ws281x adafruit-circuitpython-neopixel
sudo python3 -m pip install --force-reinstall adafruit-blinka
sudo pip3 install adafruit-circuitpython-servokit
```

Εγκαθιστούμε το βοηθητικό πρόγραμμα λήψης python wget με την εντολή:

```
sudo pip3 install wget
```

Για την εγκατάσταση του εργαλείου comitup εκτελούμε τα παρακάτω:

```
sudo apt-get install comitup
sudo mv /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/wpa.conf
sudo systemctl disable systemd-resolved
systemctl mask dnsmasq.service
```