



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

“ΕΛΕΓΧΟΣ ΚΑΤΑΣΚΕΥΗΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΡΩΤΟΤΥΠΟΥ ΜΕ
ΧΡΗΣΗ LEGO ROBOT ΚΑΙ ΓΛΩΣΣΑ ΡΥΘΜΟΝ”

ΕΛΙΝΑ ΧΑΝΤΖΑΡΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΓΓΕΛΙΔΗΣ ΠΑΝΤΕΛΗΣ

ΚΟΖΑΝΗ

ΙΟΥΛΙΟΣ 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

“ΕΛΕΓΧΟΣ ΚΑΤΑΣΚΕΥΗΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΡΩΤΟΤΥΠΟΥ ΜΕ
ΧΡΗΣΗ LEGO ROBOT ΚΑΙ ΓΛΩΣΣΑ ΡΥΘΜΩΝ”

ΕΛΙΝΑ ΧΑΝΤΖΑΡΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΓΓΕΛΙΔΗΣ ΠΑΝΤΕΛΗΣ

ΚΟΖΑΝΗ

ΙΟΥΛΙΟΣ 2020

ΠΕΡΙΛΗΨΗ

Η εκπαιδευτική ρομποτική είναι σχετικά ένας καινούργιος επιστημονικός κλάδος που ασχολείται με την κατασκευή, τον προγραμματισμό και την εφαρμογή των ρομπότ σε εκπαιδευτικό επίπεδο. Τα τελευταία χρόνια εντάσσεται ολοένα και περισσότερο στην παγκόσμια αλλά και στην ελληνική εκπαιδευτική κοινότητα. Καθίσταται αναγκαία η ύπαρξη κατάλληλα εκπαιδευμένου και τεχνολογικά καταρτισμένου εκπαιδευτικού προσωπικού που θα μπορέσει να εξελίξει την επιμόρφωση των μαθητών που βρίσκονται σε προχωρημένο στάδιο.

Το παρόν εκπαιδευτικό σενάριο απευθύνεται κυρίως στους εκπαιδευτικούς που θέλουν να εξελίξουν τον προγραμματισμό και τις κατασκευές του πακέτου Ρομποτικής Lego Mindstorms Education EV3. Η εκπαιδευτική γλώσσα EV3 έχει αντικατασταθεί από μία πραγματική γλώσσα προγραμματισμού, την Python (έκδοση python3).

Αρχικά, έχει δημιουργηθεί ένα εκπαιδευτικό υλικό μαθημάτων σε Python με βάση τα ήδη υπάρχοντα μαθήματα-κινήσεις του λογισμικού Lego Mindstorms Education EV3. Μεταρρυθμίζουμε πλέον τον εγκέφαλο του ρομπότ σε Linux με ένα νέο λειτουργικό σύστημα το εν3dev το οποίο μας ανοίγει καινούργιους ορίζοντες προγραμματισμού σε διάφορες γλώσσες.

Τα μαθήματα χωρίζονται σε δύο διαφορετικές ενότητες. Αρχικά, μαθαίνουμε να εκτελούμε προγράμματα με σύνδεση ssh, δίνοντας μας αυτό την ευκαιρία να μάθουμε βασικές γνώσεις λειτουργικού Linux. Ακόμα, γνωρίζουμε κάποια λογισμικά τα οποία μας είναι απαραίτητα (WinSCP, Putty). Η δεύτερη ενότητα αναφέρεται στο γνωστό λογισμικό Visual Studio Code το οποίο μας παρέχει ευκολότερη διασύνδεση σε ένα περιβάλλον ιδανικό και φιλικό προς τον χρήστη.

Εκτός των άλλων το Visual Studio Code μας δίνει την δυνατότητα να αναπτύξουμε περισσότερο τις γνώσεις μας αναζητώντας την κατάλληλη έκδοση γλώσσας Python που μας ταιριάζει. Συλλέγουμε πληροφορίες για την micropython και έπειτα την συγκρίνουμε με την python3. Τέλος, παρατίθενται οι μελλοντικές επεκτάσεις και τα συμπεράσματα τα οποία αποτελούν εκτός από τεκμηριωμένες προτάσεις - αποτελέσματα με βάση την πορεία της εργασίας και προσωπικές απόψεις.

ABSTRACT

Educational robotics is relatively a new discipline involved in the engineering, programming and application of robots at the educational level. In recent years, it has become increasingly integrated into both the global and the Greek educational community. It is necessary to have suitably and technologically trained teaching staff, which will be able to develop the training of advanced pupils.

This educational scenario is aimed primarily at teachers who want to develop the programming and engineering of the Lego Mindstorms Education EV3 robotics package. The educational EV3 language has been replaced by an actual programming language, Python (python version3).

Initially, a Python training course has been created based on the already existing Lego Mindstorms Education EV3 software courses. We are now switching the robot's brain to Linux with the use of a new operating system named ev3dev, which opens up new programming opportunities in various languages.

The courses are divided into two different sections. In the first section we learn how to run programs through SSH connection, giving us the opportunity to gain basic Linux knowledge. Moreover, we learn some necessary software tools (WinSCP, Putty). The second section refers to the popular Visual Studio Code software, which gives us an easier interface in an ideal and user-friendly environment.

In addition, the Visual Studio Code software enables us to further develop our knowledge by searching for the appropriate Python language version that suits us. We also collect information about micropython and then we compare it to python3. Finally we address future prospects, as well as the conclusions, which are well-documented statements resulting from research and personal views.

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους εκείνους τους καθηγητές όλα αυτά τα χρόνια στάθηκαν δίπλα μας στα δύσκολα και στα εύκολα μαθησιακά μας χρόνια καθώς επίσης και όσους βοήθησαν για την εκπόνηση της διπλωματικής μου εργασίας.

Ευχαριστώ θερμά τον επιβλέπων καθηγητή μου, κύριο Παντελή Αγγελίδη για την εμπιστοσύνη που μου έδειξε εξ' αρχής, αναθέτοντάς μου το συγκεκριμένο θέμα.

Επιπλέον ευχαριστώ τον λέκτορα του τμήματος κύριο Μηνά Δασυγένη για την παραχώρηση εξοπλισμού και φυσικά τα παιδιά της ρομποτικής ομάδας HyperionRobotics του Πανεπιστημίου, για την συμβολή τους και την άριστη προθυμία τους σε οτιδήποτε χρειαστώ.

Ευχαριστώ πολύ την αγαπητή συνάδελφο αλλά και ξεχωριστή μου φίλη, Κατερίνα Βαφειάδου, για την στήριξή της όλα αυτά τα χρόνια σε όλα τα επίπεδα.

Ιδιαίτερες ακόμα ευχαριστίες στον άνθρωπο που με έφερε κοντά στην ρομποτική, που με τις πολύτιμες συμβουλές του και το αμείωτο ενδιαφέρον του για όλη την ομάδα, μας έκανε μόνο όμορφα συναισθήματα να έχουμε αποκομίσει για την ρομποτική. Αναφέρομαι στον κύριο Δημήτρη Παύλου, κάτοχο του φροντιστηρίου ArtiRobots, στο οποίο πραγματοποίησα την πρακτική μου και εργάστηκα.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου που με πίστεψε και με στήριξε αληθινά όλα αυτά τα χρόνια.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΕΙΣΑΓΩΓΗ	9
1.1 Ορισμός του ρομπότ.....	9
1.2 Ρομποτική και ιστορική αναδρομή	10
1.3 Ρομποτική και Τεχνητή Νοημοσύνη.....	14
1.4 Τα ρομπότ στην υπηρεσία του ανθρώπου	15
1.5 Εκπαιδευτική ρομποτική και τα οφέλη της.....	18
1.6 Εκπαιδευτική ρομποτική και προγραμματισμός.....	19
LEGO MINDSTORMS.....	21
2.1 Lego Robots	21
2.2 Λογισμικό Lego Mindstorms EV3 Education και εκπαιδευτικά οφέλη	22
2.3 Δομικά στοιχεία πακέτου Lego Education Mindstorms EV3	23
2.4 Αισθητήρες πακέτου Lego Mindstorms EV3	25
2.5 Κινητήρες και εγκέφαλος	27
ΣΥΣΤΗΜΑΤΑ ΔΙΕΚΠΕΡΑΙΩΣΗΣ ΚΩΔΙΚΑ.....	29
ΕΙΣΑΓΩΓΗ	29
3.1 Βοήθεια από το λογισμικό Mindstorms EV3 Education.....	30
3.2 Τα πρώτα Λειτουργικά Συστήματα	31
3.2.1 Λειτουργικό σύστημα EV3DEV	32
3.2.2 SSH και Putty.....	33
3.2.3 WinSCP.....	35
3.2.4 Βασικές εντολές πλοήγησης σε λειτουργικό Linux	38
3.2.5 Εντολές πλοήγησης στο Putty και σύνδεση με ev3dev.....	40
3.3 Λογισμικό Visual Studio Code.....	42
3.3.1 Σύνδεση EV3DEV στο Visual Studio Code.....	42
ΡΥΘΘΟΝ ΚΑΙ MICROPΥΘΘΟΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ VSC.....	44
4.1 Η γλώσσα προγραμματισμού Python	44

4.2	Η έκδοση python3 για κατασκευές EV3DEV	44
4.3	Η έκδοση MicroPython	48
4.4	Σύγκριση εκδόσεων python3 και MicroPython.....	51
4.5	Μελλοντικές επεκτάσεις.....	54
4.6	Συμπεράσματα	55
ΠΑΡΑΡΤΗΜΑ		57
.....		57
Εκπαιδευτικά μαθήματα σε python3 και Putty		57
[i]	Κώδικας Ευθεία Κίνηση – Straight Move.....	57
[ii]	Κώδικας Περιστροφή – Move Tank.....	59
[iii]	Κώδικας Stop σε αντικείμενο – Stop at Object.....	60
Εκπαιδευτικά μαθήματα σε python3 και Visual Studio Code.....		61
[i]	Κώδικας Μετακίνηση Αντικειμένου – Move Object	61
[ii]	Κώδικας Καμπυλωτή Κίνηση – Curved Move	62
[iii]	Κώδικας Stop στη γραμμή – Stop at Line.....	63
[iv]	Κώδικας Stop σε συγκεκριμένη γωνία – Stop at Angle	64
ΒΙΒΛΙΟΓΡΑΦΙΑ		66

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Ένα από τα ρομπότ - χελώνες που κατασκεύασε ο W.Grey Walter	12
Εικόνα 2 Το Shakey στον ειδικά σχεδιασμένο χώρο του	13
Εικόνα 3 Ένα ακόμα από τα πρώιμα ρομπότ, το HILARE	13
Εικόνα 4 Δομικά στοιχεία πακέτου Lego Education Mindstorms EV3.....	23
Εικόνα 5 Διαφορετικοί τύποι γκραναζιών	24
Εικόνα 6 Απεικόνιση αισθητήρα χρώματος.....	25
Εικόνα 7 Απεικόνιση γυροσκοπικού αισθητήρα.....	26
Εικόνα 8 Απεικόνιση αισθητήρα απόστασης.....	26
Εικόνα 9 Απεικόνιση αισθητήρα αφής	27
Εικόνα 10 Απεικόνιση κινητήρα.....	27
Εικόνα 11 Απεικόνιση εγκεφάλου	28
Εικόνα 12 Πλοήγηση στο λογισμικό Mindstorms Education EV3.....	30
Εικόνα 13 Υποδοχή κάρτας SD	33
Εικόνα 14 Σύνδεση εν3dev στο Putty	34
Εικόνα 15 Login και password για σύνδεση στο εν3dev	35
Εικόνα 16 Σύνδεση εν3dev στο WinSCP	37
Εικόνα 17 Περιβάλλον WinSCP	37
Εικόνα 18 Εντολές πλοήγησης σε λειτουργικό Linux.....	41
Εικόνα 19 Πρόγραμμα σε ρython3 από προσωπικό αρχείο εκπαιδευτικών μαθημάτων	45
Εικόνα 20 Πρόγραμμα σε ρython3 από προσωπικό αρχείο εκπαιδευτικών μαθημάτων	47
Εικόνα 21 Δείγμα γραφής κώδικα σε MicroPython.....	49
Εικόνα 22 Κομμάτι κώδικα για ήχο	50
Εικόνα 23 Κομμάτι κώδικα Αισθητήρα - 2 κινητήρων	50
Εικόνα 24 Σύγκριση Python3 - MicroPython.....	51

ΕΙΣΑΓΩΓΗ

1.1 Ορισμός του ρομπότ

Ο ορισμός της λέξης ρομπότ εξελίχθηκε με την πάροδο του χρόνου, παράλληλα με τα άλματα της έρευνας και την πρόοδο της τεχνολογίας.

Έναν αιώνα πριν καθιερώθηκε για πρώτη φορά με την σημερινή του έννοια ο όρος ρομπότ από τον Τσέχο θεατρικό συγγραφέα Κάρελ Τσάπεκ στο έργο του Rossum's Universal Robots. Τα περισσότερα λεξικά αναφέρουν ως δημιουργό της λέξης “ρομπότ” τον Capek, αλλά πιο ανεπίσημες πηγές ισχυρίζονται ότι εκείνος που επινόησε τον όρο ήταν ο αδερφός του, ο Josef. Σε κάθε περίπτωση, η λέξη ρομπότ προέρχεται από την τσέχικη λέξη *robota* (ρομπότα) που σημαίνει καταναγκαστική εργασία και *robotnik* που σημαίνει δουλοπάροικος. Το 1920 λοιπόν, το έργο αυτό, σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) της τεχνολογικής εξέλιξης. Σε πολλές σύγχρονες σλαβικές γλώσσες (π.χ. την πολωνική) χρησιμοποιείται σαν έκφραση της καθημερινότητας με την έννοια της σκληρής δουλειάς. Τα περισσότερα ρομπότ σήμερα πράγματι χρησιμοποιούνται για την εκτέλεση καταναγκαστικής εργασίας με τη μορφή επαναλαμβανόμενων ενεργειών. Παρόλα αυτά η ρομποτική είναι πολλά υποσχόμενη. [1]

Με την πρόοδο της επιστήμης και της τεχνολογίας η έννοια της λέξης ρομπότ έγινε πιο σύνθετη και θα μπορούσε να εκφραστεί από τον παρακάτω ορισμό: "ρομπότ είναι μια επαναπρογραμματιζόμενη πολυλειτουργική χειριστική διάταξη, σχεδιασμένη για τη μετακίνηση υλικών, εξαρτημάτων, εργαλείων και εξειδικευμένων διατάξεων, μέσω μεταβλητών, προγραμματισμένων κινήσεων για την εκτέλεση μιας σειράς εργασιών". Θα μπορούσαμε να πούμε ότι ένα ρομπότ συγκροτείται από δύο συστήματα, το μηχανικό στο οποίο περιλαμβάνεται το σύστημα της κίνησης και στο ηλεκτρονικό στο οποίο εμπίπτει και η επαναπρογραμματιζόμενη μνήμη του. Υπάρχουν διάφορα κριτήρια διαφοροποίησης και αντίστοιχες κατηγοριοποιήσεις των ρομπότ. Μία από αυτές είναι η διάκρισή τους σε τρεις "γενιές". Στην πρώτη γενιά ταξινομούνται ρομπότ με ελάχιστη ευκαμψία, που διευθύνονται από τον άνθρωπο όπως, για παράδειγμα, οι απλοί "χειριστές", που επιτρέπουν τη μετακίνηση επικίνδυνων αντικειμένων όπως ραδιενεργών υλικών. Στη δεύτερη γενιά ταξινομούνται τα ρομπότ που είναι εφοδιασμένα με σταθερό πρόγραμμα δράσης και ρομπότ που λαμβάνουν

εντολές από κάποιο σύστημα αριθμητικού ελέγχου. Στην τρίτη γενιά ταξινομούνται ρομπότ που είναι εφοδιασμένα:

- ✓ με αισθητήριες "πληροφορίες" από το περιβάλλον.
- ✓ με διάταξη επεξεργασίας των πληροφοριών.
- ✓ με κινητήριο σύστημα εκτέλεσης εργασιών. [2]

1.2 Ρομποτική και ιστορική αναδρομή

Η ρομποτική είναι η μελέτη των ρομπότ, δηλαδή η μελέτη της αυτοδύναμης και σκοπούμενης άποψης και δράσης τους στο φυσικό κόσμο. Λέγεται ότι ο ιδιαίτερα παραγωγικός συγγραφέας επιστημονικής φαντασίας Isaac Asimov ήταν ο πρώτος που χρησιμοποίησε τον όρο ρομποτική, με βάση τον όρο ρομπότ του Capek. Αν αυτό αληθεύει τότε ο Asimov είναι αυτός που ονομάτισε οργανωμένα αυτή τη μεγάλη και ταχέως αναπτυσσόμενη επιστημονική και τεχνολογική ανάπτυξη.

Το πρώτο ρομπότ που κατασκευάστηκε ποτέ είναι η Χελώνα του W. Grey Walter. Τα τρία πεδία που είναι σημαντικός παράγοντας στην ιστορία της ρομποτικής, από το ταπεινό της ξεκίνημα ως τα σημερινά επιτεύγματα που αποτελούν την αιχμή του δόρατος της τεχνολογίας είναι:

- ✓ Η θεωρία ελέγχου.
- ✓ Η κυβερνητική.
- ✓ Η τεχνητή νοημοσύνη.

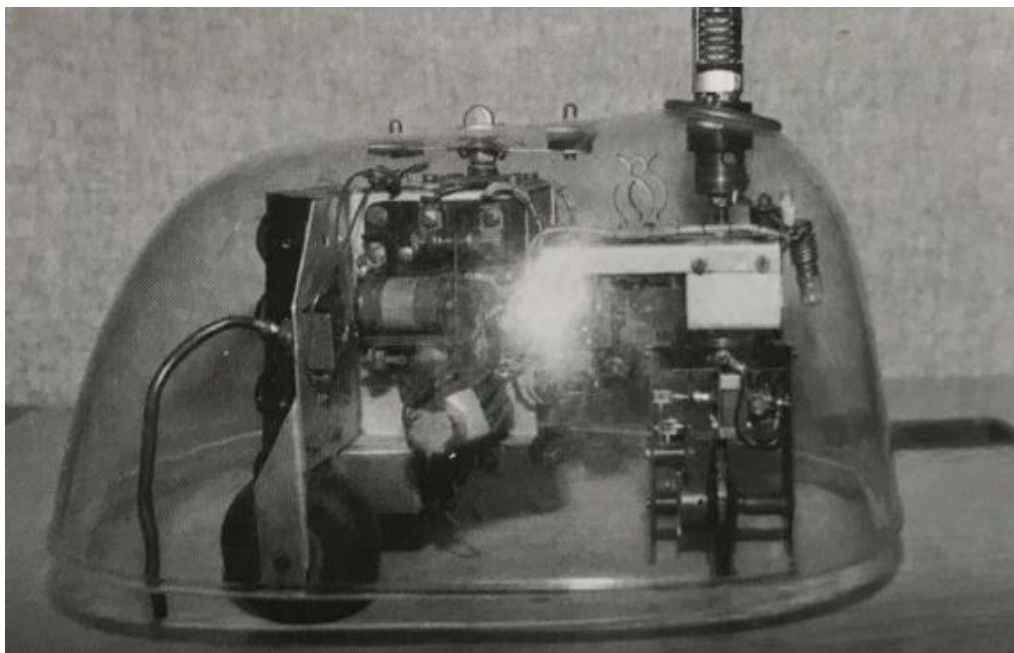
Η θεωρία ελέγχου είναι η μαθηματική έρευνα των ιδιοτήτων που χαρακτηρίζουν τα αυτοματοποιημένα συστήματα ελέγχου, από τις ατμομηχανές έως τα σύγχρονα αεροσκάφη. Τόσο η τέχνη της εκτέλεσης αυτοματοποιημένων συστημάτων όσο και η επιστήμη που ερευνά τον τρόπο λειτουργίας, έχουν τις ρίζες τους από τα αρχαία χρόνια. Μελετήθηκε εκτενώς ως τμήμα της επιστήμης των μηχανολόγων μηχανικών που ασχολείται με τον σχεδιασμό, την κατασκευή και τις φυσικές ιδιότητες των μηχανών. Παράλληλα χρησιμοποιήθηκε για την μελέτη των υδραυλικών συστημάτων, έπειτα των συστημάτων ελέγχου θερμοκρασίας και των ανεμόμυλων και τελικά των ατμομηχανών που οδήγησαν στην βιομηχανική επανάσταση. Μέχρι τις αρχές του 20^{ου} αιώνα, τα κλασικά μαθηματικά εφαρμόστηκαν για την τυποποίηση και την κατανόηση αυτών των συστημάτων και τελικά κατέληξαν στη σημερινή μορφή θεωρίας ελέγχου.

Όσο η θεωρία ελέγχου αναπτυσσόταν και ωρίμαζε, στα χρόνια του Β' παγκοσμίου πολέμου δημιουργήθηκε ένα καινούργιο ερευνητικό πεδίο, η Κυβερνητική. Οι ευεργητές του, μελέτησαν τα βιολογικά συστήματα από το επίπεδο των νευρώνων έως το επίπεδο των συμπεριφορών τους. Επιπλέον, επιδίωξαν να εφαρμόσουν παρόμοιες αρχές σε απλά ρομπότ, χρησιμοποιώντας τις μεθόδους της θεωρίας ελέγχου. Η κυβερνητική λοιπόν, ασχολούνταν με την έρευνα και τη συγκρισιμότητα θεμάτων επικοινωνίας και διεργασιών ελέγχου σε βιολογικά και τεχνητά συστήματα. Ο William ήταν ένας καινοτόμος νευροφυσιολόγος που ασχολήθηκε με τον τρόπο λειτουργίας του εγκεφάλου. Κατά τη διάρκεια της έρευνάς του, κάπου στη δεκαετία του 1940 και ύστερα, κατασκεύασε πλήθος νοήμων σχεδιασμένων μηχανών τις οποίες ονόμασε χελώνες. Οι χελώνες του ήταν απλά ρομπότ με τρεις τροχούς εγκατεστημένους όπως στα τρίκυκλα: έναν τροχό μπροστά για ρύθμιση της κατεύθυνσης και δύο τροχούς πίσω για κίνηση. Καλύπτονταν από ένα πλαστικό διάφανο κέλυφος, το οποίο αναπαριστούσε αληθοφανή εικόνα, τουλάχιστον στα μάτια ενός φιλικού και ανεπηρέαστου παρατηρητή. Τα τμήματα που αποτελούνταν, μία από τις χελώνες του, εν ονόματι Machina Specylatrix, ήταν:

- ✓ Ένα φωτοκύτταρο
- ✓ Έναν αισθητήρα επαφής
- ✓ Μία επαναφορτιζόμενη μπαταρία
- ✓ Τρεις κινητήρες για κάθε τροχό
- ✓ Τρεις τροχούς
- ✓ Και ένα αναλογικό ηλεκτρονικό κύκλωμα που χρησιμοποιούσε δύο ηλεκτρονικές λυχνίες και λειτουργούσε ως εγκέφαλος συνδέοντας τους δύο αισθητήρες με τροχούς.

Έτσι λοιπόν ο William απέδωσε στην μηχανή του τις εξής ικανότητες:

- ✓ Εντοπισμός του φωτός
- ✓ Κίνησης προς το φως
- ✓ Απομάκρυνσης από το φως πολύ υψηλής έντασης
- ✓ Στροφής και απόθησης για της αποφυγή εμποδίων
- ✓ Και επαναφόρτισης των μπαταριών

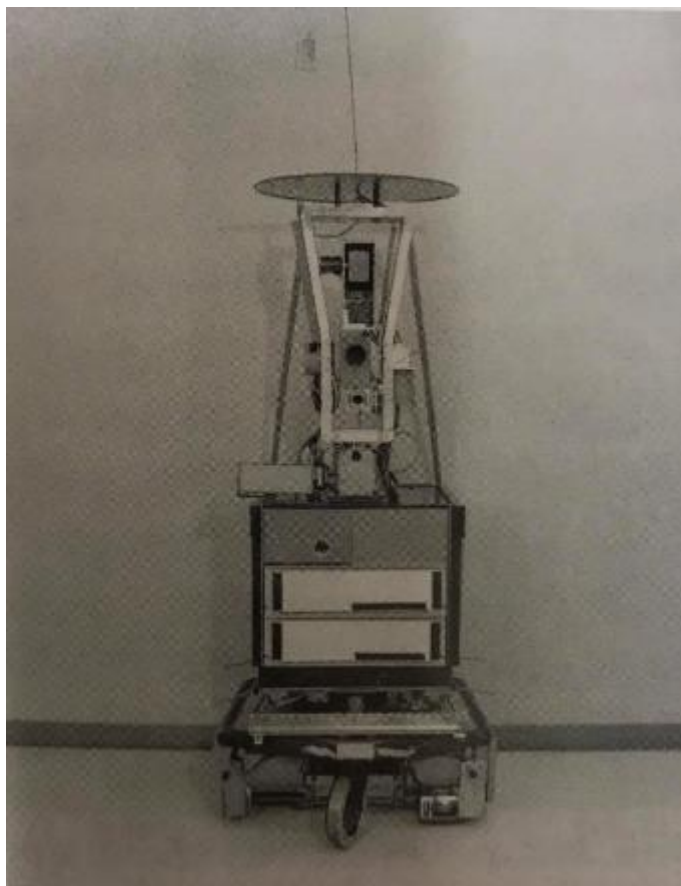


Εικόνα 1 Ένα από τα ρομπότ - χελώνες που κατασκεύασε ο W.Grey Walter

Τέλος, ο κλάδος της Τεχνητής Νοημοσύνης είναι το τρίο πεδίο το οποίο ήρθε στο φως επίσημα το 1956 σε ένα συνέδριο που διεκπεραιώθηκε στο πανεπιστήμιο του Hanover. Στην συνάντηση αυτή στην οποία συμμετείχαν οι πιο ξακουστοί ερευνητές της τότε εποχής, συζήτησαν για τη δημιουργία νοημοσύνης στις μηχανές. Τα συμπεράσματα περιληπτικά ήταν τα εξής: Για να θεωρούνται οι μηχανές ευφυείς θα πρέπει να έχουν την δυνατότητα να πραγματοποιούν περίπλοκες νοητικές διεργασίες χρησιμοποιώντας τα ακόλουθα:

- ✓ Εσωτερικά μοντέλα του κόσμου
- ✓ Αναζήτηση μεταξύ των πιθανών λύσεων
- ✓ Σχεδιασμός ενεργειών και συλλογιστική για την επίλυση προβλημάτων
- ✓ Ιεραρχική οργάνωση συστήματος
- ✓ Ακολουθιακή εκτέλεση προγραμμάτων [3]

Το πιο σημαντικό όμως αποτέλεσμα της συνάντησης αυτής ήταν η θεμελίωση και η διαμόρφωση του κλάδου της ρομποτικής βασισμένη στην τεχνητή νοημοσύνη (AI-inspired robotics), Πρόκειται για ένα νέο πεδίο το οποίο επικεντρωνόταν στη συλλογιστική. Το πρώτο ρομπότ βασισμένο στην τεχνητή νοημοσύνη κατασκευάστηκε στα τέλη της δεκαετίας του 1960 στο ινστιτούτο του Palo Alto στην California και διέθετε αισθητήρες επαφής και μία κάμερα. Ονομάστηκε Shakey και βρισκόταν σε ένα δωμάτιο με λευκό δάπεδο και μερικά μεγάλα μαύρα αντικείμενα, όπως μπάλες και πυραμίδες. Όταν μετακινούταν σε αυτό το δωμάτιο δημιουργούσε πλάνα αργά και προσεκτικά. Το πιο κατάλληλο για αυτόν ήταν οι άνθρωποι να μην μπλέκονται στα πόδια του. Το σώμα του τρανταζόταν καθώς κινούνταν για να εκτελέσει τα πλάνα, εξού και η ονομασία του Shakey.



Εικόνα 2 Το Shakey στον ειδικά σχεδιασμένο χώρο του

Ένα ακόμα ρομπότ που αναπτύχθηκε πάνω στο ίδιο πεδίο της τεχνητής νοημοσύνης ρομποτικής είναι το HILARE. Δημιουργήθηκε στα τέλη της δεκαετίας του 1970 στο εργαστήριο ανάλυσης και αρχιτεκτονικής συστημάτων στην Toulouse της Γαλλίας. Περιλάμβανε μία βιντεοκάμερα, αισθητήρες υπερήχων και ένα αποστασιόμετρο λέιζερ. Το συγκεκριμένο ρομπότ ήταν αντικείμενο μελέτης από πολλές γενιές ερευνητών και αποτελεί ένα από τα μακροβιότερα ρομπότ έως και τώρα.



Εικόνα 3 Ένα ακόμα από τα πρώιμα ρομπότ, το HILARE

1.3 Ρομποτική και Τεχνητή Νοημοσύνη

Όπως ήδη αναφέραμε η ρομποτική είναι ένας σύγχρονος και πρόσφορος τεχνολογικός κλάδος της αυτοματοποίησης ο οποίος ασχολείται με τη λειτουργία των ρομπότ, τη μελέτη, τον σχεδιασμό και την έρευνα για την περιπλέον ανέλιξή τους. Πιο περιληπτικά, μελετά τις μηχανές που θα είναι σε θέση να αντικαταστήσουν τον άνθρωπο στην εκτέλεση μίας εργασίας, η οποία συνδυάζει τη φυσική δραστηριότητα με τη διαδικασία λήψης αποφάσεων.

Αναφέραμε παραπάνω τις τρεις κατηγορίες γενεών των ρομπότ. Αξίζει να σχολιάσουμε ότι τα ρομπότ της πρώτης γενιάς δεν είχαν ικανότητα εκτίμησης και αίσθησης. Τα ρομπότ της 2^{ης} γενιάς ασκούν πτωτική υπολογιστική ικανότητα ενώ τα ρομπότ της 3^{ης} γενιάς διαθέτουν «νοημοσύνη» και μπορούν να φέρουν προς αντιμετώπιση προβλήματα αλλά και να παίρνουν αποφάσεις κατά τη διάρκεια της εργασίας τους. Τις ικανότητες αυτές αποκτούν με τεχνικές «τεχνητής νοημοσύνης» και «αίσθησης». [4] Έτσι λοιπόν αρχίζει να γενιέται ένα μεγάλο κεφάλαιο της ρομποτικής η λεγόμενη Τεχνητή Νοημοσύνη η οποία αναφέρεται στον κλάδο της πληροφορικής και ο οποίος ασχολείται με την σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς. Όλα αυτά υπονοούν μια νοημοσύνη η οποία συνεπάγεται με μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, κατανόηση, αντιμετώπιση προβλημάτων και δεξιότητες που πρέπει να χρησιμοποιεί ένα ρομπότ τρίτης γενιάς.

Παρόλο που η Τεχνητή Νοημοσύνη, ως ανεξάρτητο πεδίο μελέτης, είναι σχετικά καιρία, οι καταβολές της είναι αρκετά πιο παλιές. Η αλήθεια είναι ότι ξεκίνησε πριν από 2.400 χρόνια, όταν ο Έλληνας φιλόσοφος Αριστοτέλης εφηύρε της έννοια της λογικής συλλογιστικής (logical reasoning). Η δοκιμή ολοκλήρωσης της γλώσσας της λογικής συνεχίστηκε με τον Leibniz και τον Νεύτωνα. Τον 19^ο αιώνα ο George Boole ανέπτυξε την άλγεβρα Μπουλ, η οποία έθεσε τα θεμέλια των υπολογιστικών κυκλωμάτων. Παρόλα αυτά, η κεντρική ιδέα της νοήμων μηχανής προήρθε από τον Alan Turing. Ο όρος Τεχνητή Νοημοσύνη χρησιμοποιήθηκε για πρώτη φορά από τον John McCarthy το 1956. [5]

Η τεχνητή νοημοσύνη αποτελεί σημείο τομής μεταξύ πολλών επιστημών όπως της πληροφορικής, της ψυχολογίας, της φιλοσοφίας, της νευρολογίας, της γλωσσολογίας και της επιστήμης μηχανικών, με σκοπό τη σύνθεση νοήμων συμπεριφοράς, με στοιχεία συλλογιστικής, μάθησης και προσαρμογής στο περιβάλλον ενώ παράλληλα δεν είναι λίγες οι φορές που εφαρμόζεται σε μηχανές ή υπολογιστές ειδικής κατασκευής. Χωρίζεται σε δύο κατηγορίες, στη συμβολική τεχνητή νοημοσύνη και στην υποσυμβολική. Η πρώτη προσπαθεί να εξομοιώσει την ανθρώπινη νοημοσύνη αλγοριθμικά χρησιμοποιώντας σύμβολα και λογικούς κανόνες υψηλού επιπέδου. Αυτό λοιπόν είναι ένα κομμάτι στο οποίο θα σταθούμε σε αυτήν την εργασία, «τον προγραμματισμό» και συγκεκριμένα σε Python.

1.4 Τα ρομπότ στην υπηρεσία του ανθρώπου

Το 1942 ο Asimov διατύπωσε τους ακόλουθους τρεις νόμους της ρομποτικής στην ιστορία Runaround.

- ✓ Ένα ρομπότ δεν επιτρέπεται ενεργητικά ή λόγω απραξίας του να βλάψει ένα ανθρώπινο ον.
- ✓ Ένα ρομπότ πρέπει να υπακούει ένα ανθρώπινο ον, εκτός αν οι εντολές που δέχεται αντιτίθενται στον πρώτο νόμο.
- ✓ Ένα ρομπότ πρέπει να προστατεύει την ύπαρξή του, αρκεί αυτό να μην αντιτίθεται στον πρώτο ή δεύτερο νόμο.

Οι παραπάνω αναφερόμενοι λοιπόν νόμοι έχουν στόχο την εφαρμογή στην εξυπηρέτηση των αναγκών του ανθρώπου, με βάση τις κινηματογραφικές ταινίες και τα έργα επιστημονικής φαντασίας που αποτέλεσαν το μέσο για την επίτευξη αυτών.

Ως πρώτο στοιχειώδες γεγονός είναι ότι τα ρομπότ λαμβάνουν εντολές μόνο από τον κατασκευαστή. Στην σύγχρονη εποχή τα ρομπότ εκτελούν πλέον περισσότερες εργασίες και δραστηριότητες, επιπροσθέτως σε μεγάλο βαθμό στην αλληλεπίδραση με τους ανθρώπους. Ως αποτέλεσμα τα ρομπότ την σήμερα ημέρα αποκτούν νέο σημαντικό ρόλο στην κάλυψη σύγχρονων συνεχών αναγκών στην κοινωνία. Ανάγκες στην καθημερινότητα του ανθρώπου, όπως στην εφαρμογή παροχής υπηρεσιών και στην ψυχαγωγία, μέχρι την αυτοματοποίηση μηχανημάτων στα εργοστάσια και στον τομέα της ιατρικής περίθαλψης

Τα ρομπότ στο Διάστημα:

Είναι γνωστό ότι στην εξερεύνηση των πλανητών του ηλιακού μας συστήματος χρησιμοποιήθηκαν ρομποτικές διαστημικές συσκευές. Τροχιακοί δορυφόροι, συσκευές προσεδάφισης και εξερεύνησης εδάφους είναι κάποιες από τις συσκευές που έχουν επισκεφθεί την σελήνη, τους κομήτες, τους δορυφόρους, αστεροειδής και πλανήτες. Κάποιοι από τους επιστήμονες στηρίζουν την άποψη ότι τα ρομπότ μπορούν να ανταπεξέλθουν στο διάστημα και να αντικαταστήσουν την ανθρώπινη ύπαρξη χωρίς το ρίσκο της ζωής του ανθρώπου και μάλιστα με λιγότερο κόστος. Ο άνθρωπος ως ταξιδιώτης στο διάστημα απαιτεί λεπτομερή συστήματα για την υποστήριξή του. Παρόλο την εξέλιξη της τεχνολογίας στις σημερινές εποχές απαιτείται πολύς χρόνος για τον άνθρωπο ως προς την επίτευξη πρόωσης σε οποιονδήποτε άλλον προορισμό εκτός της σελήνης. Σύμφωνα με τους ειδικούς οι άνθρωποι και τα ρομπότ μπορούν να επιβιώσουν σε πολλών ετών ταξίδια με σκοπό να φέρουν εις πέρας την αποστολή εξερεύνησης. Πάρα ταύτα μια μεγάλη μερίδα επιστημών διαφωνούν και θεωρούν ότι οι άνθρωποι είναι αναντικατάστατοι σε τέτοιου είδους αποστολές, διότι τα ρομπότ είναι προγραμματισμένα σε συγκεκριμένες εντολές. Αντίθετα οι άνθρωποι έχουν την ικανότητα να αναγνωρίσουν και να εντοπίσουν άγνωστα στοιχεία που μπορεί να προκύψουν σε σύγκριση με τα ρομπότ που έχουν συγκεκριμένες βάσεις πληροφοριών με αποτέλεσμα την ανικανότητα αντιμετώπισης προκείμενων άγνωστων στοιχείων. Οπότε σε σύγκριση με τα

ρομπότ οι άνθρωποι σε περίπτωση κάποιου απρόβλεπτου τεχνικού προβλήματος μπορούν να ανταπεξέλθουν και να το επισκευάσουν.

Τα ρομπότ και οι άνθρωποι προσφέρουν διαφορετικές αλλά εξίσου σημαντικές πληροφορίες και δυνατότητες στην μελέτη και εξερεύνηση του Διαστήματος, οπότε στόχος είναι η συνεργασία μεταξύ τους. Στόχος αυτής της συνεργασίας δηλαδή είναι τα ρομπότ να εξυπηρετούν τον άνθρωπο σε περιπτώσεις και εργασίες που είναι επικίνδυνες και μη εφικτές για τον ίδιο. Από την άλλη ο άνθρωπος θα έχει τον ρόλο του να μπορεί να αντιδράσει καλύτερα στο απροσδόκητο και να αποσαφηνίζει νέες εντολές που προκύπτουν για να ανταπεξέλθει το ρομπότ. Όπως αναφέρει και ο Jim Bell, αστρονόμος και πλανητικός επιστήμονας στο Πανεπιστήμιο του Cornell:

“Αν και οι αποστολές αστροναυτών είναι πολύ πιο δαπανηρές και επικίνδυνες από τις αποστολές ρομποτικών σκαφών, είναι απολύτως ζωτικής σημασίας για την επιτυχία των προγραμμάτων εξερεύνησης.”

Τα ρομπότ στην Ιατρική:

Στην Ιατρική τα ρομπότ έχουν επεμβατικό ρόλο καθώς η βοήθειά τους είναι πολυσήμαντη στην χειρουργική και στην ορθοπεδική, εφαρμόζοντας τον ακριβή έλεγχο και την μεγάλη ευκρίνεια χρήσης των χειρουργικών εργαλείων.

Επίσης, τα ρομπότ αναφέρονται και στην αντιμετώπιση της γαστροοισοφαγικής παλινδρόμησης, όπως στις επεμβάσεις χοληδόχο κύστη με ενδοσκόπιο που εισάγεται από το σώμα.

Ο Lawrence Ho και η ομάδα του, στο Singapore's National University Hospital κατασκεύασαν και δουλεύουν πάνω στην ανάπτυξη ενός μικροσκοπικού ρομπότ που είναι μόλις μήκους τεσσάρων χιλιοστών και με την βοήθεια ενός ενδοσκοπίου εισάγεται από το στόμα κινείται μέσα στο σώμα του ανθρώπου και με την στήριξη ενός ρομποτικού μικροσκοπικού μεγέθους το οποίο μπορεί να εντοπίσει θανατηφόρες ασθένειες. Για παράδειγμα, απομακρύνει καρκινικό ιστό και καυτηριάζει αιμοφόρα αγγεία. Με τα λεγόμενα του Ho το μικροσκοπικό αυτό ρομπότ θα κάνει πολύ πιο εύκολη την θεραπεία του καρκίνου του στομάχου, αφού αποτελεί τη δεύτερη συχνότερη αιτία θανάτου από καρκίνο σε παγκόσμιο επίπεδο.

Καθώς τα ρομπότ με την πάροδο του χρόνου γίνονται ολοένα και μικρότερα, οι προγραμματιστές θέτουν στόχους να τα εξοπλίζουν με τεχνητή νοημοσύνη. Αυτή η τεχνολογία είναι απαραίτητη εφόσον η ιατρική κοινότητα στοχεύει στην θεραπεία των ασθενών, τη βελτίωση της ποιότητας ζωής και την άμεση πρόληψη των προβλημάτων υγείας.

Παρόλα αυτά η ρομποτική δεν μπορεί να αντικαταστήσει εξ ολοκλήρου την παρουσία ενός εξειδικευμένου και σκεπτόμενου χειρουργού. Η ρομποτική και κατ'επέκταση η τεχνολογία μπορεί να λύσει τα χέρια του γιατρού και να του προσφέρει σπουδαία εργαλεία για την διευκόλυνσή του.

Είναι εμφανές ότι με την εξέλιξη της τεχνολογίας, τα ρομπότ θα γίνονται ολοένα και περισσότερα με ικανότητες πολυκκοκότερες με την πάροδο του χρόνου. Μία εύλογη ερώτηση βέβαια θα ήταν αν η ωφέλεια που προσφέρουν τα ρομπότ αντισταθμίζει το μεγάλο κόστος που απαιτείται για την έρευνα αλλά και την κατασκευή τους. Απάντηση δεν υπάρχει παρόλα αυτά όμως κάθε επιστημονική έρευνα και τεχνολογική εφαρμογή οφείλει να πάντοτε να προσπαθεί για τη βελτίωση της ποιότητας της ζωής του ανθρώπου.

Ρομπότ στη βιομηχανία:

Ο ερχομός του ρομπότ στη βιομηχανία απάλλαξε τον άνθρωπο από επίπονες βιομηχανικές εργασίες καθώς επίσης έπαιξε καθοριστικό ρόλο και στην μείωση της δαπανώμενης ενέργειας και κατ'επέκταση στην ελάττωση των εξόδων της βιομηχανίας.

Η πρώτη εταιρία που παρήγαγε ρομπότ ήταν πραγματικότητα το 1956. Η ρομποτική έγινε ευρέως γνωστή στα τέλη του 1970 καθώς πολλές εταιρείες των ΗΠΑ εισήλθαν στον τομέα αυτό. Το 1973 ήρθαν για πρώτη φορά ρομπότ στην βιομηχανική αγορά της Ευρώπης. Με σκοπό την διευκόλυνση της ζωής όλων μας, η βιομηχανία θα παίζει καθοριστικό ρόλο.

Θετικά:

- Λόγω της αφομοίωσης των ρομπότ στην εμπορική οικονομία, οι άνθρωποι θα απολαμβάνουν τις ίδιες ανέσεις σε ολοένα μετριάζον κόστος.
- Θα υπάρξει αύξηση στο βιομηχανικό ρυθμό παραγωγής λόγω της αυξανόμενης ταχύτητας, επακόλουθο της τυποποίησης που προσφέρει η ρομποτική.
- Ο μηχανικός χαρακτήρας των ρομπότ προσφέρει ασφάλεια καθώς, χωρίς ανθρώπινη παρουσία, εκμηδενίζονται οι πιθανότητες ανθρώπινου λάθους.
- Με την πάροδο του χρόνου, οι κατασκευαστές των ρομπότ αποκτούν οικολογική συνείδηση, κατασκευάζοντας τα ρομπότ φιλικά προς το περιβάλλον, με αποτέλεσμα να παράγεται ίδιο αποτέλεσμα με ελάχιστους ρύπους.

Αρνητικά:

- Πρόκληση ανεργίας σε πολλούς εργασιακούς τομείς.
- Οι άνθρωποι κάνουν πλέον χρήση έτοιμων, αναλώσιμων τεχνολογιών, χωρίς να θέτουν το μυαλό τους σε ουσιαστική λειτουργία, μετατρέποντας τον άνθρωπο οκνηρό, αμβλύνοντας τις γνώσεις του.
- Άτομα που δεν καταβάλλουν προσπάθεια ίσως παρουσιάσουν υγειονομικά προβλήματα είτε επιδεινώσουν τα ήδη υπάρχοντα, όπως η παχυσαρκία, καθώς ο τρόπος ζωής τους θα απλουστεύσει σημαντικά.
- Οι σημερινοί έφηβοι, όντας εξοικειωμένοι με την τεχνολογία, αποτελούν και τις πιο ευάλωτες προσωπικότητες σε αυτήν. Συχνή έκθεση στην βία και άλλους αρνητικούς παράγοντες.
- Περιβαλλοντική ρύπανση από τις καύσεις στα αέρια που χρησιμοποιούν τα ρομπότ για τη λειτουργία τους. [6]

1.5 Εκπαιδευτική ρομποτική και τα οφέλη της

Η εκπαιδευτική ρομποτική είναι σχέδιο διεπιστημονικής δραστηριότητας και περιλαμβάνει τομείς όπως η επιστήμη, τα μαθηματικά, η πληροφορική και η τεχνολογία. παραχωρώντας καινούργιες ευκαιρίες στην εκπαίδευση αλλά και σε όλα τα επίπεδα. Η εκπαιδευτική ρομποτική είναι μια ικανή και επιδέξια διδασκαλία μάθησης ενθαρρύνοντας τους μαθητευόμενους να κατασκευάσουν, να δημιουργήσουν και να ελέγξουν τα ρομπότ χρησιμοποιώντας συγκεκριμένες γλώσσες προγραμματισμού. Η εκπαιδευτική ρομποτική προάγει έναν ευχάριστο τρόπο μάθησης, και ταυτόχρονα δίνει ώθηση στα κίνητρα των παιδιών, την συνεργασία, την αυτοπεποίθηση και τη δημιουργικότητα. Πολλοί ερευνητές αναφέρουν ότι τα προγράμματα ρομποτικής δημιουργούν έναν πολύτιμο δρόμο για την διερεύνηση του ενδιαφέροντος των παιδιών και τη συμμετοχή τους στην επιστήμη, την τεχνολογία, τη μηχανική και τα μαθηματικά, και παράλληλα τα παρακινούν ώστε να ανθήσουν και να ακολουθήσουν μια σταδιοδρομία σε έναν από αυτούς τους τομείς.

Η εκπαιδευτική ρομποτική είναι σχετικά ένας καινούργιος επιστημονικός κλάδος που ασχολείται με την κατασκευή, τον προγραμματισμό και την εφαρμογή των ρομπότ σε εκπαιδευτικό επίπεδο. Συγχωνεύτηκε ως διδακτικό αντικείμενο στο μάθημα της πληροφορικής και σκοπεύει στο σχεδιασμό δραστηριοτήτων για την ενίσχυση δεξιοτήτων και ικανοτήτων υπολογιστικής σκέψης. Έχει παρατηρηθεί ότι την τελευταία δεκαετία έχει δημιουργήσει μεγάλη απήχηση στους εκπαιδευτικούς πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, λόγω της αίσιας έκβασης με την οποία σπρώχνει τους μαθητές στην απόκτηση δεξιοτήτων επίλυσης προβλημάτων. [7]

Η εκπαιδευτική ρομποτική εμπνέεται από:

- τις κονστρακτιβιστικές θεωρίες του Jean Piaget, ο οποίος θεωρεί πως η μάθηση στον άνθρωπο δεν είναι αποτέλεσμα μετάδοσης της γνώσης, αλλά μια ενεργητική διαδικασία κατασκευής της γνώσης που βασίζεται στις εμπειρίες. (Piaget, 1972)
- την κονστρακσιονιστική εκπαιδευτική φιλοσοφία του S. Papert που προσθέτει ότι η απόκτηση καινούργιας γνώσης συντελείται πιο αποτελεσματικά όταν αυτοί που μαθαίνουν ασχολούνται με την κατασκευή προϊόντων που έχουν προσωπικό νόημα για αυτούς. Σκοπός της κονστρακσιονιστικής εκπαίδευσης είναι να προσφέρει στα παιδιά κατάλληλα πράγματα να κάνουν έτσι ώστε να μάθουν στην πράξη, με πιο αποτελεσματικό τρόπο από ό, τι πριν.

Με βάση τα παραπάνω και όχι μόνο, αρχίζει να λαμβάνει χώρο μια κοινωνικο-εποικοδομητική άποψη όπου η μάθηση δεν είναι εξατομικευμένη αλλά αποτελεί κοινωνική και κοινωνικοποιημένη δραστηριότητα. Έτσι λοιπόν, η χρήση της εκπαιδευτικής ρομποτικής έχει θετικές επιπτώσεις εκτός από το γνωστικό τομέα και στο συναισθηματικό όπως:

- ✓ Αυτοεκτίμηση
- ✓ Αυτοπεποίθηση

Και στο κοινωνικό:

- ✓ Κοινωνικοποίηση

✓ Απομυθοποίηση

Η σχεδίαση, η κατασκευή και ο προγραμματισμός των ρομπότ δίνουν την ευκαιρία στα παιδιά να διευρύνουν τους ορίζοντές τους παίζοντας και να αναπτύξουν δεξιότητες. [8]

1.6 Εκπαιδευτική ρομποτική και προγραμματισμός

Η διδασκαλία του προγραμματισμού έχει αρχίσει να εξαπλώνεται σε όλα τα σχολεία παγκοσμίως. Ο πλανήτης μας εξελίσσεται, οι νέες γενιές μεγαλώνουν κοντά στην τεχνολογία, οπότε και η δημιουργία νέων τάσεων στην εκπαίδευση είναι πλέον απαραίτητη προϋπόθεση για τις απαιτήσεις της σύγχρονης εποχής.

Ο προγραμματισμός στην εποχή μας υπάρχει παντού και περιλαμβάνει ένα ευρύ φάσμα εφαρμογών. Καλείται ως το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος, υλοποιώντας κάποιους αλγορίθμους για την αυτοματοποιημένη εκτέλεση εργασιών ή την επίλυση κάποιου υπολογιστικού προβλήματος. Θεμελιώδη ρόλο στον υπολογιστικό προγραμματισμό καθορίζουν οι χιλιάδες διαφορετικές γλώσσες προγραμματισμού, δηλαδή οι προτυποποιημένες τυπικές γλώσσες οι οποίες είναι απαραίτητες για τη δημιουργία προγράμματος.

Όσον αφορά τώρα την εκπαιδευτική ρομποτική υπάρχουν κάποιες γλώσσες προγραμματισμού ειδικά σχεδιασμένες για παιδιά. Μόλις τα παιδιά εξοικειωθούν με την λογική του προγραμματισμού μπορούν εύκολα να προχωρήσουν στην εκμάθηση πραγματικών γλωσσών οι οποίες θα τους προσκομίσουν πολλά οφέλη στη μετέπειτα ζωή τους. Τα οφέλη της διδασκαλίας προγραμματισμού σε παιδιά μπορούν να ομαδοποιηθούν στις παρακάτω κατηγορίες:

Προετοιμασία για την αγορά εργασίας: Σύμφωνα με διάφορες εκτιμήσεις, η αγορά θα έχει ανάγκη από όλο και περισσότερους ειδικευόμενους σε θέματα πληροφοριών και τεχνολογίας.

“Η τεχνολογία και οι υπολογιστές βρίσκονται στο επίκεντρο της οικονομίας μας. Προκειμένου να προετοιμαστούμε για τις απαιτήσεις του 21ου αιώνα - και να επωφεληθούμε από τις ευκαιρίες - είναι σημαντικό όλο και περισσότεροι μαθητές να αρχίσουν να μαθαίνουν βασικές δεξιότητες προγραμματισμού, ανεξάρτητα από τον τομέα εργασίας που σκοπεύουν να ακολουθήσουν.”

Park Todd, Γενικός Διευθυντής Τεχνολογίας ΗΠΑ

Αναπτυξιακά και εκπαιδευτικά πλεονεκτήματα: Η εκμάθηση του προγραμματισμού δεν συνεπάγεται οφέλη μόνο όσον αφορά την τεχνολογία αλλά τις γνωστικές και κοινωνικές δεξιότητες των μαθητών. Τα παιδιά παρουσιάζουν πρόοδο σε διάφορους ακαδημαϊκούς τομείς όπως είναι η οπτική μνήμη και οι γλωσσικές δεξιότητες.

“Η εκμάθηση του προγραμματισμού κάνει τα παιδιά να αισθάνονται ενδυναμωμένα, δημιουργικά και με αυτοπεποίθηση. Εάν θέλουμε οι νέες γυναίκες να διατηρήσουν αυτά τα χαρακτηριστικά στην ενήλικη ζωή τους, μια εξαιρετική επιλογή είναι να γνωρίσουν τον προγραμματισμό υπολογιστών από μικρή ηλικία.”

Susan Wojcicki, Αντιπρόεδρος Google

Υπολογιστική σκέψη: Είναι η διαδικασία του να θέτεις λύσεις σε ένα πρόβλημα, οι οποίες λύσεις διαμορφώνονται με έναν κατανοητό τρόπο και με μία λογική σειρά. Προωθεί δεξιότητες όπως είναι η αποσύνθεση, δηλαδή το σπάσιμο των μεγάλων εργασιών σε μικρότερες, τη συλλογιστική και τη διόρθωση σφαλμάτων.

Εξέλιξη της λογικής σκέψης: Η λογική σκέψη συνδέεται άμεσα με την επιτυχή ολοκλήρωση αλγορίθμων, δηλαδή ενός προκαθορισμένου συνόλου βημάτων, τα οποία εκτελούνται για την επίτευξη του στόχου.

Μάθηση μέσα από τα λάθη: Τα λάθη είναι ένα αναπόφευκτο κομμάτι στον προγραμματισμό. Η αποσφαλμάτωση όμως αποτελεί ένα σημαντικό μέρος της δημιουργικής διαδικασίας καθώς το λάθος στον προγραμματισμό αποκτά μία εκπαιδευτική αξία πρόκλησης. [9]

LEGO MINDSTORMS

2.1 Lego Robots

Η ύπαρξη αλλά και η εξέλιξη των ρομπότ, θα μπορούσαμε να πούμε ότι ξεκίνησε την δεκαετία του 1950 και περνώντας μέσα από διάφορες γενιές εξέλιξης, φτάνει στα τέλη του 1990. Τη δεκαετία αυτή οι εταιρείες στρέφουν τα ρομπότ τους για πρώτη φορά εκτός βιομηχανικού περιβάλλοντος. Ένα από τα ρομπότ που ήρθαν στην επιφάνεια τότε ήταν και το Lego Mindstorms kit το οποίο κατασκευάστηκε το 1998. Η πρώτη έκδοση των LEGO Mindstorms κυκλοφόρησε στην αγορά με την επωνυμία Robotics Invention System η οποία αποτελείται από 717 τεμάχια τα οποία χωρίζονται σε αισθητήρες, κινητήρες, Lego τούβλα και ένα RCX προγραμματιζόμενο τούβλο που περιέχει τρεις θέσεις εισόδου και τρεις θέσεις εξόδου που συνδέεται με έναν μικροελεγκτή.

Ο πρώτος διαγωνισμός ρομποτικής πραγματοποιήθηκε στα σχολεία μέσης εκπαίδευσης τον νοέμβριο του 1998. Η ομάδα της Lego παρουσίασε τα Lego Mindstorms σε διάφορους περιφερειακούς διαγωνισμούς. Ένας πιλοτικός διαγωνισμός υπάρχει ακόμα στο μουσείο επιστήμης και βιομηχανίας του Σικάγο, όπου οι συμμετέχοντες ομάδες ανέρχονταν τις 200.

Μετά την πρώτη γενιά της Lego Mindstorms kit ήρθε να πάρει την θέση της η Lego Mindstoms NXT το 2006. Το τούβλο NXT στον πυρήνα του συστήματος είναι ένας ελεγκτής πολλαπλών χρήσεων ο οποίος μπορεί γρήγορα και εύκολα να συνθεθεί με έναν ηλεκτρονικό υπολογιστή. Πρόκειται για έναν κύριο επεξεργαστή του NXT στα 32bit Atmel ARM, με 256KB flash memory και μνήμη RAM 64KB. Διαθέτει 4 κουμπιά διεπαφής και μία οθόνη LCD διαστάσεων 100x64 pixels. Υπάρχει ακόμα καλώδιο θύρας USB και Bluetooth για επικοινωνία με τον υπολογιστή. Το Lego Mindstorms kit περιλαμβάνει 577 τεμάχια εκ των οποίων οι τρεις είναι σερβομηχανές, οι τέσσερις αισθητήρες ήχου, φωτός, αφής αλλά και υπερήχων. Επιπρόσθετα, το πακέτο περιλαμβάνει ένα γραφικό περιβάλλον, το NXT-G, όπου πρόκειται για ένα λογισμικό το οποίο εγκαθίσταται στον υπολογιστή και μέσω του οποίου ο χρήστης μπορεί να συντάξει πρόγραμμα ενώ μπορεί και απευθείας να πραγματοποιήσει τη λήψη του στο NXT.

Η αμέσως επόμενη γενιά έρχεται τον Ιανουάριο του 2013, με την Lego Mindstorms EV3. Ο πυρήνας του Lego Mindstorms EV3 αποτελεί το προγραμματιζόμενο μέρος του ο οποίος διαθέτει τέσσερις θύρες εξόδου για τους κινητήρες και τέσσερις θύρες εισόδου για τους αισθητήρες. Πλέον μπορεί να υπάρξει σύνδεση με τον υπολογιστή μέσω USB, Bluetooth ή και με ασύρματη σύνδεση WiFi. Διαθέτει ακόμα LCD οθόνη καθώς και 16MB flash memory και μνήμη RAM 64MB. Τέλος, υπάρχει επιλογή προσθήκης κάρτας εσωτερικής μνήμης SD Card με μέγεθος έως και 32GB. [10]

2.2 Λογισμικό Lego Mindstorms EV3 Education και εκπαιδευτικά οφέλη

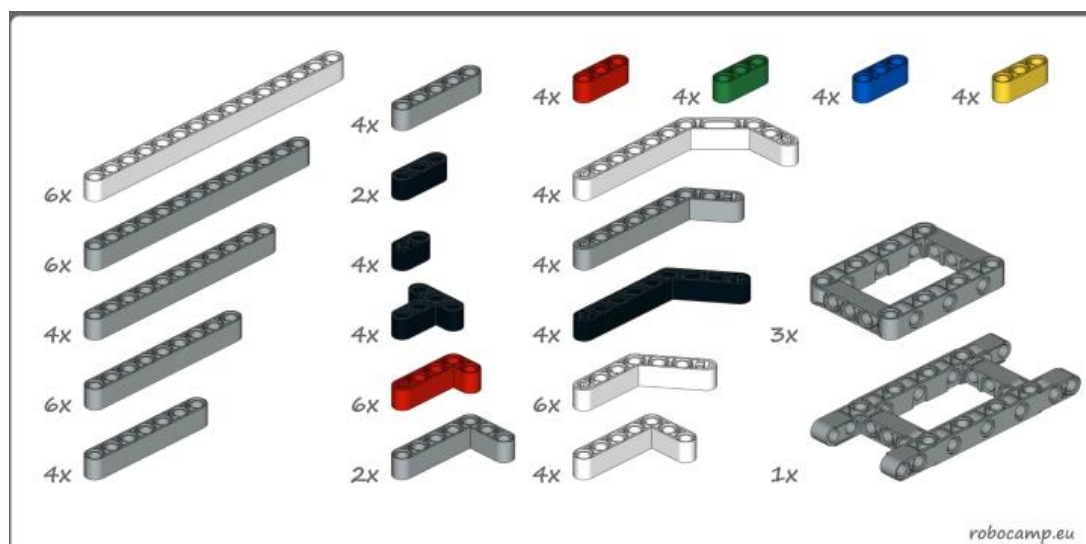
- Το LEGO MINDSTORMS Education EV3 είναι κατάλληλα διαμορφωμένο για εκπαίδευση στην τάξη και περιέχει ό,τι χρειάζεστε για να διδάξετε μαθήματα εκπαιδευτικής ρομποτικής.
- Δίνει την ευκαιρία στους μαθητές να κατασκευάσουν, να προγραμματίσουν και να πειραματιστούν με τις δικές τους λύσεις σε πραγματικά προβλήματα της ρομποτικής τεχνολογίας. Το πακέτο LEGO MINDSTORMS Education EV3 περιέχει το EV3 Intelligent Brick, το οποίο είναι ένας μικρός υπολογιστής που κάνει εφικτό τον έλεγχο μοτέρ και την συλλογή δεδομένων από τους αισθητήρες.
- Επιτρέπει Bluetooth και Wi-Fi επικοινωνία και παρέχει συλλογή δεδομένων και προγραμματισμό.
- Δίνεται κίνητρο στους μαθητές να σκεφτούν ώστε να μπορέσουν να βρουν δημιουργικές λύσεις σε προβλήματα αλλά και να τις αναπτύξουν κατά τη διαδικασία επιλογής, κατασκευής, δοκιμής και αξιολόγησης.
- Δίνεται η ευκαιρία οι μαθητές να επικοινωνήσουν μεταξύ τους, να συνεργαστούν και να αποκτήσουν εμπειρία και δεξιότητες κατά τη χρήση μιας σειράς αισθητήρων, κινητήρων και έξυπνων μονάδων.
- Είναι συμβατό με τους αισθητήρες Hi-Technic.

Εκπαιδευτικά οφέλη:

- ✓ Σχεδιασμός και κατασκευή προγραμματιζόμενων ρομπότ με την χρήση μοτέρ, αισθητήρων, τροχών, αξόνων και άλλων τεχνικών εξαρτημάτων.
- ✓ Κατανόηση δισδιάστατων σχεδίων με στόχο την κατασκευή τρισδιάστατων σχεδίων.
- ✓ Κατασκευή, έλεγχος, διόρθωση και ανακατασκευή σχεδίων για την βελτίωση της απόδοσης του ρομπότ.
- ✓ Η απόκτηση ουσιαστικής εμπειρίας με την χρήση μαθηματικών ιδεών όπως η εκτίμηση και η μέτρηση απόστασης, χρόνου και ταχύτητας. [11]

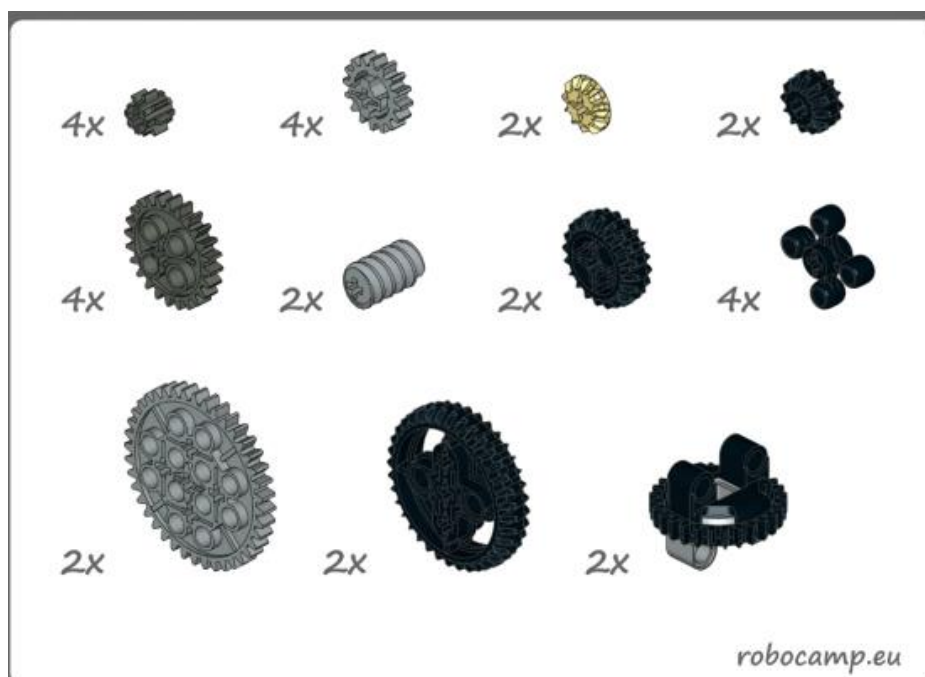
2.3 Δομικά στοιχεία πακέτου Lego Education Mindstorms EV3

Το λευκό και το γκρι είναι τα χρώματα που κατά βάση κυριαρχούν και στα 541 τουβλάκια του πακέτου Education, αλλά υπάρχουν και κάποιες πινελιές κόκκινου, μαύρου και κάποιων άλλων αποχρώσεων. Θα δείτε δοκούς Technic σε 8 διαφορετικά μήκη, 7 τύπους γωνιακών δοκών και δύο είδη πλαίσιων. Ακόμα θα βρείτε δοκούς με 3 οπές σε 5 διαφορετικά χρώματα (κόκκινο, πράσινο, μπλε, κίτρινο, μαύρο), έτσι ώστε να είστε σε θέση να τα χρησιμοποιήσετε για πειράματα με τον αισθητήρα χρώματος. Το πακέτο περιέχει ακόμα άξονες με 10 διαφορετικά μήκη και 3 τύπους αξόνων με πεπλατυσμένη τη μία άκρη.



Εικόνα 4 Δομικά στοιχεία πακέτου Lego Education Mindstorms EV3

Για τη σύνδεση των αξόνων με τις δοκούς και τα πλαίσια, μπορούμε να χρησιμοποιήσουμε 7 τύπους πύρων (στους οποίους περιλαμβάνονται πύροι με ή χωρίς τριβή για δημιουργία διαφόρων συνδέσμων) ή έναν από τους 18 διαθέσιμους τύπους συνδέσμων, συμπεριλαμβανομένων και συνδέσμων από άξονα σε πύρο. Για να ενώσουμε ή να διαχωρίσουμε δοκούς πάνω σε έναν άξονα, μπορούμε να επιλέξουμε ανάμεσα σε 2 είδη δακτυλίων ή και έναν σύνδεσμο από πύρο σε πύρο (γνωστός και ως σωλήνας). Επιπρόσθετα, το εκπαιδευτικό πακέτο περιλαμβάνει και 10 διαφορετικούς τύπους γραναζιών (για κάθετη ή σε σειρά σύνδεση) και δύο μικρές περιστρεφόμενες βάσεις. Συνολικά, ιφύστανται 30 στοιχεία για τη δημιουργία μηχανισμών με γρανάζια. Για τον λόγο αυτό, μπορούμε να σχεδιάσουμε μοντέλα με διαφορετικές μεταδόσεις κίνησης και σχέσεις μετάδοσης.



Εικόνα 5 Διαφορετικοί τύποι γραναζιών

Εάν θέλουμε να δημιουργήσουμε ένα όχημα, μπορούμε να χρησιμοποιήσετε τους δύο μεγάλους τροχούς με τα φαρδιά ελαστικά, ή τους μικρότερους τροχούς με τους ιμάντες ή και τη μικρή μεταλλική μπίλια που περικλείεται από μία πλαστική θήκη, η οποία είναι και ένα από τα πιο σημαντικά κομμάτια του πακέτου. Για παράδειγμα, μπορεί να χρησιμοποιηθεί ως ένας μεγάλος τροχός στήριξης στο πίσω μέρος του ρομπότ για κίνηση με δύο ρόδες. Οι ιμάντες που περιλαμβάνονται στο σετ είναι κατασκευασμένοι από πολλαπλά πλαστικά κομμάτια τα οποία συνδέονται μεταξύ τους και σε συνδυασμό με τους αντίστοιχους τροχούς μπορούμε να δημιουργήσουμε έναν λειτουργικό ιμάντα σε διάφορα μεγέθη. Ακόμα μπορούμε να κατασκευάσουμε ερπύστριες οι οποίες δουλεύουν αρκετά καλά. Λόγω όμως του ότι η πλαστική επιφάνεια είναι ομαλή, υπάρχει περίπτωση να δημιουργηθεί και κάποια ολίσθηση. Η λύση σε αυτό το πρόβλημα βρίσκεται στην αγορά μικρών στοιχείων από καουτσούκ τα οποία αν συνδεθούν στις οπές, αυξάνουν πολύ σημαντικά την πρόσφυση. Τα στοιχεία αυτά είναι διαθέσιμα στο πακέτο extension. Παρόλα αυτά, μπορούμε πάντα να αγοράσουμε όλα αυτά τα στοιχεία ξεχωριστά. [12]

Επίσης περιλαμβάνει:

- 3 σερβο-μοτέρ με ενσωματωμένους αισθητήρες περιστροφής.
- Αισθητήρα χρώματος, γυροσκόπιο, αισθητήρα υπερήχων (απόστασης) και 2 αισθητήρες αφής.
- Τροχό με σφαιρίδιο.
- Επαναφορτιζόμενη μπαταρία DC.
- Καλώδια σύνδεσης.
- Οδηγίες κατασκευής .
- Άδεια λογισμικού προγραμματισμού (Single Licence Software).

2.4 Αισθητήρες πακέτου Lego Mindstorms EV3

Οι αισθητήρες τους οποίους περιλαμβάνει το πακέτο Lego Education Mindstorms EV3 όπου είναι και η τελευταία έκδοση είναι οι εξής:

- ✓ **Αισθητήρας χρώματος:** Ο Color Sensor έχει την ικανότητα να ανιχνεύει έως και επτά χρώματα (μαύρο, λευκό, μπλε, πράσινο, κόκκινο, κίτρινο και καφέ), αλλά και την απουσία χρώματος. Είναι ικανό ακόμα να ανιχνεύει το φως που εισέρχεται σε έναν χώρο αλλά και την έντασή του. Ο αισθητήρας χρώματος χρησιμοποιεί μία κλίμακα έντασης για το φως όπου στο 0 βρίσκεται το πολύ σκοτεινό και στο 100 το πολύ φωτεινό. Αυτό δείχνει ότι ένα ρομπότ με τον συγκεκριμένο αισθητήρα θα μπορούσε να προγραμματιστεί ώστε να κινείται σε μία λευκή επιφάνεια, μέχρι να ανιχνεύσει μία γραμμή μαύρη.



Εικόνα 6 Απεικόνιση αισθητήρα χρώματος

- ✓ **Gyro αισθητήρας:** Ο γυροσκοπικός αισθητήρας χρησιμοποιείται αρκετά συχνά για βοηθητική χρήση για την σταθερότητα του ρομπότ. Το μηχανικό κόμμάτι όμως είναι ικανό να καθορίσει τον προσανατολισμό περιστροφής του ρομπότ σε διάφορους άξονες. Παρέχει επίσης τη δυνατότητα στους χρήστες να μετρούν τις γωνίες και να σχεδιάσουν το ιδανικό σύστημα πλοήγησης για το ρομπότ τους. Είναι σημαντικό να πούμε, ότι μέσω των χαρακτηριστικών του αισθητήρα Gyro, μπορούμε να έχουμε μεγάλη ακρίβεια στις κινήσεις.



Εικόνα 7 Απεικόνιση γυροσκοπικού αισθητήρα

- ✓ **Αισθητήρας απόστασης:** Ένας αισθητήρας απόστασης ή αλλιώς Ultrasonic Sensor χρησιμοποιείται για την μέτρηση της απόστασης μεταξύ του ρομπότ και των αντικειμένων που μπορεί να συναντήσει στο περιβάλλον του. Ο αισθητήρας απόστασης λειτουργεί με τρόπο εκπομπής ηχητικών κυμάτων για την ανίχνευση και την μέτρηση της απόστασης ενός ή περισσότερων αντικειμένων. Επιπρόσθετα, ο συγκεκριμένος αισθητήρας έχει σχεδιαστεί για ένα ευρύτερο φάσμα εφαρμογών, συμπεριλαμβανομένης της μέτρησης της απόστασης και της ανίχνευσης της οποιαδήποτε κίνησης γύρω από τον χώρο που κινείται. Ο αισθητήρας απόστασης είναι ικανός να βελτιστοποιεί το ρομπότ έτσι ώστε να καταφέρνει να αποφεύγει αντικείμενα μέσα στον χώρο, να ανιχνεύει έναν κινούμενο στόχο ή κάποιον εισβολέα ή να χτυπάει συναγερμό σε περίπτωση που ένα αντικείμενο πλησιάσει ολοένα και πιο κοντά προς το μέρος του. [13]



Εικόνα 8 Απεικόνιση αισθητήρα απόστασης

- ✓ **2 αισθητήρες αφής:** Οι αισθητήρες αφής ή αλλιώς Touch Sensors είναι ένα αναλογικό και απλό εργαλείο χρήσιμο για ανίχνευση με μεγάλη ακρίβεια. Μπροστά στον αισθητήρα μπορούμε να παρατηρήσουμε ένα κουμπί το οποίο χρησιμεύει για τη μέτρηση πίεσης δίνοντας έτσι την πληροφορία ότι το ρομπότ έχει έρθει σε επαφή με κάποιο αντικείμενο που βρίσκεται στο περιβάλλον του.



Εικόνα 9 Απεικόνιση αισθητήρα αφής

2.5 Κινητήρες και εγκέφαλος

Στο πακέτο LEGO EDUCATION MINDSTORMS EV3 μπορούμε να βρούμε δύο μεγάλους κινητήρες. Όλοι οι κινητήρες έχουν ενσωματωμένο έναν αισθητήρα περιστροφής. Μία από τις πιο χρήσιμες λειτουργίες των μεγάλων κινητήρων είναι για την κίνηση ενός οχήματος με δύο ρόδες, ώστε το ρομπότ να προχωρά απλά προς τα εμπρός και πίσω αλλά και να μπορεί να στρίβει. Ο ενσωματωμένος αισθητήρας περιστροφής μας δίνει την επιλογή να χρησιμοποιήσουμε τους αισθητήρες για διάφορους λόγους. Ένα απλό παράδειγμα αναφέρεται στον μοχλό ελέγχου. Ο έλεγχος περιστροφής και στους δύο τύπους κινητήρων είναι πολύ ακριβής, με ακρίβεια 1° , χάρη στο ενσωματωμένο ταχύμετρο.

Οι μεγάλοι κινητήρες έχουν αρκετά μεγαλύτερη ισχύ και μεγάλη κεφαλή περιστροφής, γεγονός που βοηθάει στη μετάδοση κίνησης. Μπορούν να φτάσουν τις 170rpm. Ο μικρότερος κινητήρας έχει μεγαλύτερη ταχύτητα έως και 250 rpm εις βάρος όμως της διαθέσιμης ισχύος. Λόγω του μεγέθους του, μπορεί γρήγορα να ενσωματωθεί σε πολλές κατασκευές. [14]



Εικόνα 10 Απεικόνιση κινητήρα

Η πιο σημαντική λειτουργία του εγκεφάλου EV3 έχει να κάνει με τον έλεγχο των ρομπότ που έχουμε κατασκευάσει. Ο εγκέφαλος είναι στην πραγματικότητα ένας μικροελεγκτής με έναν επεξεργαστή ARM9 με λειτουργικό σύστημα Linux που είναι κατασκευασμένο σε ένα μεγάλο τούβλο LEGO. Είναι εφοδιασμένο με οθόνη, ηχείο, 6 κουμπιά (των οποίων τα 5 μπορούν να προγραμματιστούν και επίσης φωτίζονται), 4 θύρες εισόδου, 4 θύρες εξόδου, μία μίνι θύρα USB για σύνδεση με τον υπολογιστή, θύρα USB, θύρα Micro SD έως και 32 GB και μία θήκη μπαταριών. Η επικοινωνία με τη συσκευή είναι εύκολη μέσω καλωδίου USB, ενσωματωμένης μονάδας Bluetooth ή και WiFi.



Εικόνα 11 Απεικόνιση εγκεφάλου

Ο εγκέφαλος λειτουργεί πολύ καλά. Ωστόσο, το μενού του δεν είναι και πολύ φιλικό προς το χρήστη και η εκκίνηση διαρκεί κάπου 30 δευτερόλεπτα. Με το πέρας του χρόνου έρχονται στην επιφάνεια επιπλέον προβλήματα. Ο εγκέφαλος μπορεί να σβήνει ή να απενεργοποιείται από μόνος του. Σε αυτές τις περιπτώσεις, αρκετά συχνά βοηθά η αναβάθμιση του framework ή η επανεκκίνηση του εγκεφάλου.

Ο εγκέφαλος EV3 δέχεται 6 μπαταρίες AA ή ακόμα θα μπορούσε κάποιος να χρησιμοποιήσει την ειδική επαναφορτιζόμενη μπαταρία της LEGO. Η μπαταρία συμπεριλαμβάνεται στο εκπαιδευτικό πακέτο, αλλά ο φορτιστής όχι. Αξίζει επίσης να σημειωθεί ότι η μπαταρία LEGO μεγαλώνει λίγο τον εγκέφαλο, επομένως μπορεί να μας δυσκολέψει σε κάποιες κατασκευές.

ΣΥΣΤΗΜΑΤΑ ΔΙΕΚΠΕΡΑΙΩΣΗΣ ΚΩΔΙΚΑ

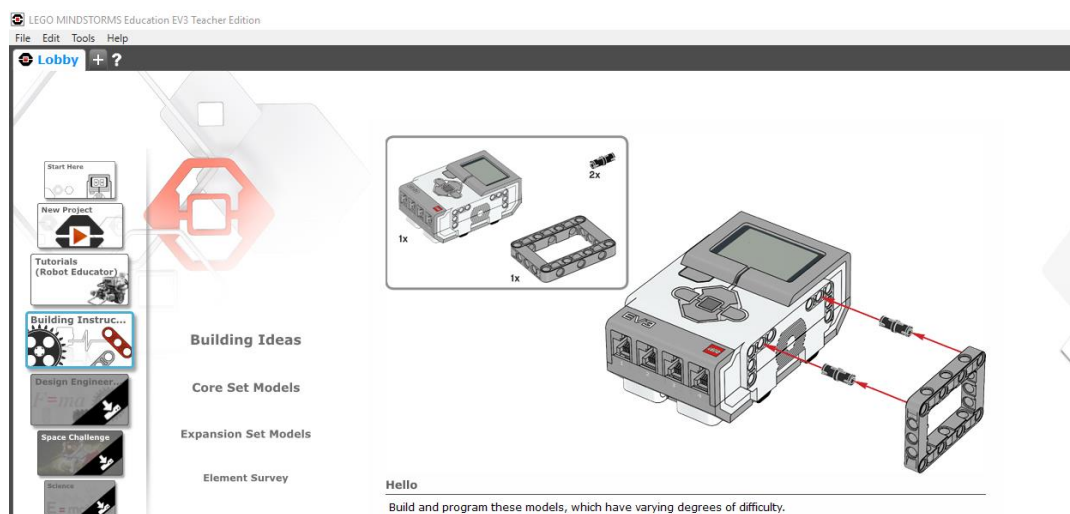
ΕΙΣΑΓΩΓΗ

Σε αυτή την ενότητα θα μιλήσουμε για δύο διαφορετικά συστήματα υλοποίησης κώδικα σε Python. Σαν πρώτο τρόπο θα χρησιμοποιήσουμε το λογισμικό Putty. Ο λόγος είναι για να μπουν τα παιδιά σε μία διαδικασία κατανόησης του λειτουργικού συστήματος Linux και για την εξοικείωσή τους με τις εντολές του συγκεκριμένου λειτουργικού συστήματος. Επιπρόσθετα, θα τους δοθεί η ευκαιρία να εγκαταστήσουν σημαντικά προγράμματα όπως αυτά που θα αναλυθούν παρακάτω τα οποία θα τα έχουν υπόψην και στην πορεία τους. Το Putty θα τους δώσει σημαντικά εφόδια στην μετέπειτα επαγγελματική τους σταδιοδρομία αν και εφόσον ακολουθήσουν τον κλάδο της πληροφορικής.

Ο δεύτερος τρόπος διεκπεραίωσης κώδικα είναι σε ένα περιβάλλον το οποίο μας τα προσφέρει όλα. Αναφέρομαι στο λογισμικό Visual Studio Code το οποίο είναι ειδικά διαμορφωμένο ώστε να υπάρχει συνδεσιμότητα με τη συσκευή EV3 μέσω User Interface. Πρόκειται για ένα φιλικό και κατανοητό προς τον χρήστη περιβάλλον που εύκολα μπορούμε να συνδέσουμε το ρομπότ μας, να εισάγουμε κώδικα Python από οποιοδήποτε αρχείο ή ακόμα και να πληκτρολογήσουμε απευθείας και να το τρέξουμε. Η εναλλαγή από τα δύσκολα στα εύκολα θα δώσει την ευκαιρία στα παιδιά να συνειδητοποιήσουν πόσα πράγματα μπορούν επιπλέον να μάθουν ακολουθώντας ένα μονοπάτι που δεν είναι όλα συμβατά μέσω ενός έτοιμου λογισμικού. Από την άλλη πρόκειται για ένα λογισμικό που επιτρέπει έναν εύκολο τρόπο συνδεσιμότητας έτσι ώστε να υπάρξει μεγαλύτερη έμφαση του κώδικα προγραμματισμού.

3.1 Βοήθεια από το λογισμικό Mindstorms EV3 Education

Η πρώτη βασική βοήθεια που μπορεί να μας προσφέρει το λογισμικό MINDSTORMS EDUCATION EV3 είναι ότι μας δίνει οδηγίες βήμα προς βήμα για να υλοποιήσουμε το μηχανικό κομμάτι της κατασκευής του ρομπότ μας. Από την καρτέλα Building Instructions μπορούμε να επιλέξουμε οποιοδήποτε μοντέλο κατασκευής μας ενδιαφέρει.



Εικόνα 12 Πλοήγηση στο λογισμικό Mindstorms Education EV3

Επιπρόσθετα, στην καρτέλα Tutorials και στη συνέχεια Basics, υπάρχουν κάποιες βασικές κινήσεις που μπορεί να υλοποιήσει το ρομπότ μας, με τον κατάλληλο βέβαια προγραμματισμό, μέσα από βιντεάκια υπόδειξης των κινήσεων. Τα 7 βασικά μαθήματα-κινήσεις που μας υποδεικνύει το λογισμικό LEGO EDUCATION MINDSTORMS EV3 είναι:

- ✓ Ευθεία Κίνηση
- ✓ Περιστροφή
- ✓ Stop σε αντικείμενο
- ✓ Μετακίνηση αντικειμένου
- ✓ Καμπυλωτή Κίνηση
- ✓ Stop στη γραμμή
- ✓ Stop σε συγκεκριμένη γωνία

Είναι σημαντικό να θυμήσουμε ότι ο προγραμματισμός σε αυτό το λογισμικό γίνεται σε μία γλώσσα ειδικά φτιαγμένη για παιδιά, την EV3. Η διαφορά είναι ότι εμείς θα ασχοληθούμε να μεν με τις ίδιες κινήσεις-μαθήματα αλλά με διαφορετική γλώσσα. Η γλώσσα την οποία θα χρησιμοποιήσουμε για να υλοποιήσουμε όλα τα παραπάνω είναι η Python και τα περιβάλλοντα στα οποία μπορούμε να εκτελέσουμε κώδικα θα αναφερθούν παρακάτω.

3.2 Τα πρώτα Λειτουργικά Συστήματα

Οι πρώτοι υπολογιστές είχαν την ικανότητα να εκτελούν μόνο μία εργασία ή ενέργεια τη φορά. Αυτός ο τρόπος λειτουργίας μπορεί να ονομασθεί ως μαζική επεξεργασία σε περιβάλλον ενός χρήστη. Ο υπολογιστής είναι σε θέση να εκτελεί μόνο ένα πρόγραμμα κάθε φορά και να επεξεργάζεται τα δεδομένα ομαδοποιημένα. Τις περισσότερες φορές, οι χρήστες σ' αυτά τα πρώτα συστήματα μπορούσαν να στείλουν τις εργασίες τους σε ένα κέντρο υπολογιστών με τη μορφή "δεσμίδων", δηλαδή διάτρητων καρτών. Για τον λόγο αυτόν, ήταν υποχρεωμένοι να αναμένουν αρκετές ώρες για να επιστρέψουν οι εκτυπώσεις των αποτελεσμάτων στα γραφεία τους.

Τα λειτουργικά συστήματα αναπτύχθηκαν με σκοπό να κάνουν πιο γρήγορη και αποτελεσματική τη χρήση των υπολογιστών. Τα πρώτα λειτουργικά συστήματα έκαναν την εναλλαγή μεταξύ εργασιών με στόχο να καταφέρουν να αυξήσουν τον όγκο της εργασίας που μπορούσαν να φέρουν εις πέρας οι υπολογιστές.

Όσο οι υπολογιστές γίνονταν ολοένα και πιο ισχυροί, έγινε σημαντικά εμφανές ότι η μαζική επεξεργασία σε περιβάλλον ενός χρήστη ήταν μη αποτελεσματική, διότι εξελίχθηκε σε πολύ χρονοβόρα αναμένοντας τις αργές συσκευές εισόδου και εξόδου να φέρουν εις πέρας τις διεργασίες τους. Έτσι ήρθε η εποχή που πολλές εργασίες ή ενέργειες θα είναι σε θέση να διαμοιράζονται τους πόρους του υπολογιστή με σκοπό την επιτυχή και γρηγορότερη αξιοποίηση του. Το γεγονός αυτό επιτυγχάνεται με τον πολυπρογραμματισμό. Ο πολυπρογραμματισμός σκοπεύει στην ταυτόχρονη λειτουργία πολλών εργασιών μαζί, οι οποίες αντιτάσσονται μεταξύ τους για την χρήση των πόρων του υπολογιστή. Στα πρώτα λειτουργικά συστήματα με τη χρήση πολυπρογραμματισμού, οι χρήστες συνέχιζαν να στέλνουν τις εργασίες τους με τον ίδιο τρόπο στον υπολογιστή με τη χρήση δηλαδή δεσμίδων και περίμεναν στην αναμονή ώρες ή μέρες μέχρι να λάβουν τα αποτελέσματα.

Στη δεκαετία του 60, πολλές ομάδες ανθρώπων, τόσο από τον τομέα των επιχειρήσεων όσο και από τον ακαδημαϊκό τομέα, ανέπτυξαν δεξιότητες και ικανότητες διαχείρισης λειτουργικών συστημάτων με δυνατότητες διαμοιρασμού χρόνου. Ο διαμοιρασμός χρόνου είναι μία διαφορετική περίπτωση προγραμματισμού κατά την οποία οι χρήστες προσεγγίζουν τον υπολογιστή μέσω τερματικών τα οποία ήταν συνήθως συσκευές με πληκτρολόγια και οθόνη. Δεκάδες και εκατοντάδες χρήστες μοιράζονται τον υπολογιστή την ίδια στιγμή. Η αλήθεια όμως είναι ότι ο υπολογιστής δεν εκτελεί την ίδια χρονικά στιγμή τα προγράμματα όλων των χρηστών. Αντί για αυτό, εκτελεί ένα μικρό τμήμα από την εργασία ενός χρήστη και ύστερα ξεκινά την εξυπηρέτηση του επόμενου χρήστη, παραχωρώντας πιθανώς τις υπηρεσίες του σε κάθε χρήστη πολλές φορές το δευτερόλεπτο. Έτσι λοιπόν, τα προγράμματα των χρηστών φαίνεται σαν να εκτελούνται την ίδια χρονικά στιγμή. Ένα μεγάλο προτέρημα του διαμοιρασμού χρόνου είναι ότι οι αιτήσεις των χρηστών λαμβάνουν άμεσα απαντήσεις από τον υπολογιστή. [15]

3.2.1 Λειτουργικό σύστημα EV3DEV

Λειτουργικό σύστημα (Operating System ή OS) ονομάζεται το λογισμικό του υπολογιστή που ευθύνεται για την εκτέλεση και τον συντονισμό των εργασιών, καθώς επίσης και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει στους χρήστες ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, μέσω του οποίου οι εφαρμογές διαπιστώνουν την εμμέσως ύπαρξη του υπολογιστή. Μια βασική αρμοδιότητα του λειτουργικού συστήματος είναι η διαχείριση του υλικού, διευκολύνοντας έτσι το λογισμικό του χρήστη από τον άμεσο και δύσκολο χειρισμό του υπολογιστή κάνοντας ευκολότερο τον προγραμματισμό τους. [16]

Όπως ήδη γνωρίζουμε, ο εγκέφαλος ενός πακέτου LEGO MINDSTORMS EDUCATION EV3 είναι σχεδιασμένος για να δουλεύει πάνω σε μία συγκεκριμένη γλώσσα, την EV3. Εφόσον, εμείς θέλουμε να προγραμματίσουμε σε Python είναι απαραίτητο να φορτώσουμε ένα καινούργιο λειτουργικό σύστημα στον εγκέφαλό μας (τον ev3dev) ο οποίος μας δίνει πολλές εναλλακτικές επιλογής γλώσσας προγραμματισμού.

Το ev3dev είναι ένα λειτουργικό σύστημα βασισμένο στο Linux που λειτουργεί με το LEGO MINDSTORMS EV3. Το ev3dev σας επιτρέπει να προγραμματίσετε σε διαφορετικές γλώσσες (π.χ. Python, C ++). Σημαντικό είναι να αναφερθεί ότι μπορείτε να γράψετε τα προγράμματα σε οποιοδήποτε IDE και όχι απευθείας μέσα στο EV3. Για να κατεβάσετε την τελευταία έκδοση του EV3dev για το EV3 πρέπει να ακολουθήσετε τον σύνδεσμο: <https://www.ev3dev.org/downloads/cs>

Καλό επίσης είναι να κατεβάσουμε και το ev3dev-stretch beta καθώς αυτό είναι απαραίτητο για τη χρήση του IDE. Οι λόγοι πιο αναλυτικά είναι οι εξής:

- Το ev3dev-stretch μας βοηθάει στο να μπορούμε να σταματήσουμε το πρόγραμμα πιέζοντας το Back bottom. Χρησιμοποιείται στο VS Code με έτοιμα πακέτα για χρήση σε γλώσσα Python (π.χ στροφή).
- Επίσης, αυτόματα παρακολουθώντας την διαφορά τάσης μπαταρίας απενεργοποιεί το EV3 όταν το επίπεδο της μπαταρίας είναι πολύ χαμηλό.

Για να πραγματοποιηθεί με επιτυχία η διαδικασία πρέπει να περάσουμε τον εικονικό δίσκο του EV3Dev στην SD κάρτα. Μπορούμε ακόμα να κατεβάσουμε και να εγκαταστήσουμε το Etcher: <https://etcher.io>. Εφόσον το εγκαταστήσουμε ανοίγουμε το πρόγραμμα και επιλέγουμε το αρχείο του EV3dev. Εισάγουμε την κάρτα μνήμης SD στον υπολογιστή και επιλέγουμε την εγγραφή στην κάρτα μνήμης. Ύστερα, την φορτώνουμε στον εγκεφαλό μας. [17]



Εικόνα 13 Υποδοχή κάρτας SD

3.2.2 SSH και Putty

Το **SSH** (Secure Shell) είναι ένα ασφαλές δικτυακό πρωτόκολλο μέσω του οποίου διεκπεραιώνεται με επιτυχία η μεταφορά δεδομένων μεταξύ δύο υπολογιστών. Το SSH είναι σε θέση να κρυπτογραφεί δεδομένα που ανταλλάσσονται κατά τη συνεδρία και ταυτόχρονα να παραχωρεί ένα ασφαλές και υπεύθυνο σύστημα αναγνώρισης καθώς και άλλες ιδιότητες όπως η ασφαλή μεταφορά αρχείων, κρυπτογραφημένο μηχανισμό για σύνδεση, εκτέλεση εντολών (SSH File Transfer Protocol) κλπ. [18]

Το SSH χρησιμοποιεί συνήθως την θύρα 22 για την σύνδεση ανάμεσα σε δύο υπολογιστές ή συσκευή στο Internet ή στο τοπικό σας δίκτυο. Χρησιμοποιείται πιο συχνά από τους διαχειριστές του δικτύου ως ένας απομακρυσμένος τρόπος ελέγχου για τη διαχείριση των διακομιστών των επιχειρήσεων τους.

Η χρήση του SSH γίνεται συνήθως με διάφορα προγράμματα, εδώ θα σας δείξουμε πως με το Putty (ένα ελεύθερο πρόγραμμα) μπορούμε να έχουμε μια ασφαλή σύνδεση ανάμεσα στον υπολογιστή μας που έχει Windows και μία συσκευή.

Αντίστοιχα, η εφαρμογή PUTTY μας παραχωρεί ένα περιβάλλον σύνδεσης και διαχείρισης ενός απομακρυσμένου λειτουργικού που υποστηρίζει το πρωτόκολλο SSH, όπως για παράδειγμα ένα Unix λειτουργικό σύστημα. Με το SSH πρωτόκολλο, δημιουργούμε ένα

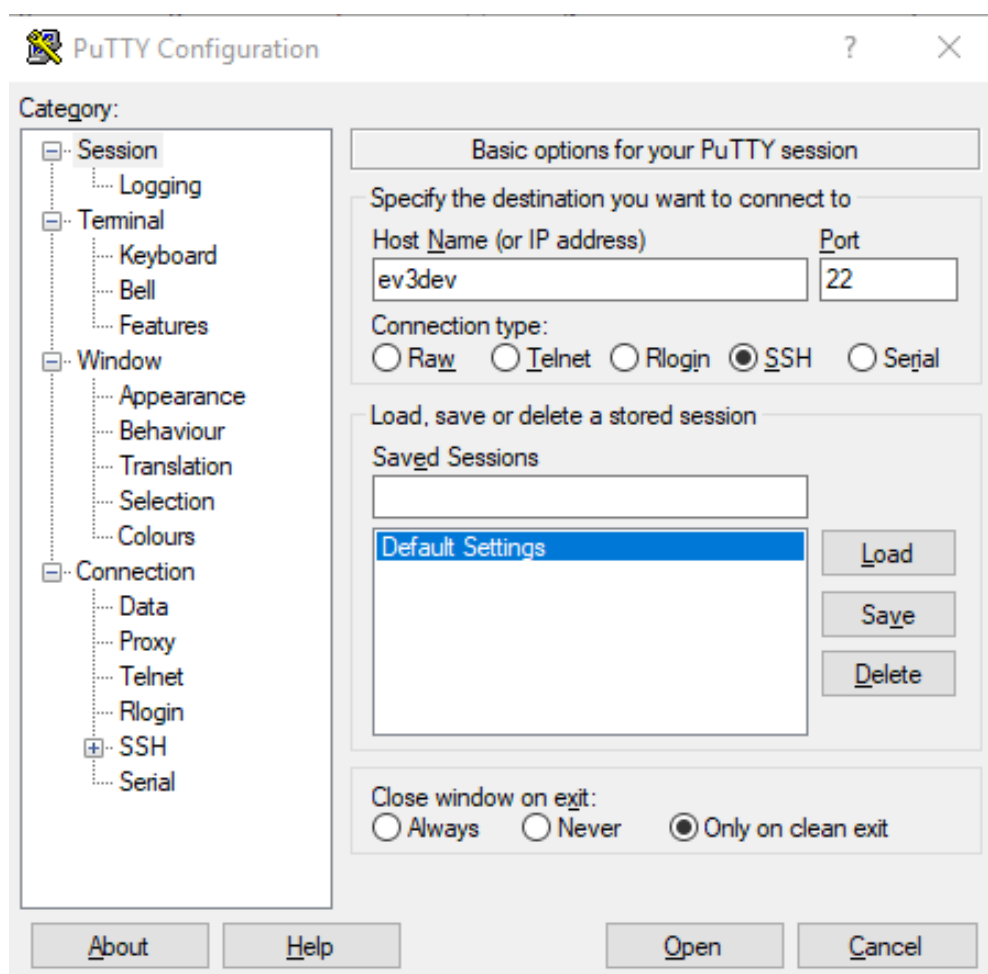
ασφαλές κανάλι επικοινωνίας μεταξύ εμάς και του απομακρυσμένου λειτουργικού συστήματος έτσι ώστε κατά την διάρκεια της σύνδεσής μας, κανείς κακόβουλος ενδιάμεσος να μην μπορεί να χρησιμοποιεί δεδομένα που ανταλλάσσουμε με το απομακρυσμένο σύστημα.

Για δική σας χρήση, μπορείτε να κατεβάσετε την εφαρμογή «**putty.exe**» ελεύθερα από την επίσημη ιστοσελίδα του.

Παρακάτω περιγράφονται τα πιο σημαντικά βήματα που χρειάζεται να ακολουθήσει κανείς για να δημιουργήσει μία καινούργια σύνδεση σε κάποιο απομακρυσμένο σύστημα και τον τρόπο με τον οποίο θα αποθηκευτούν οι ρυθμίσεις αυτές.

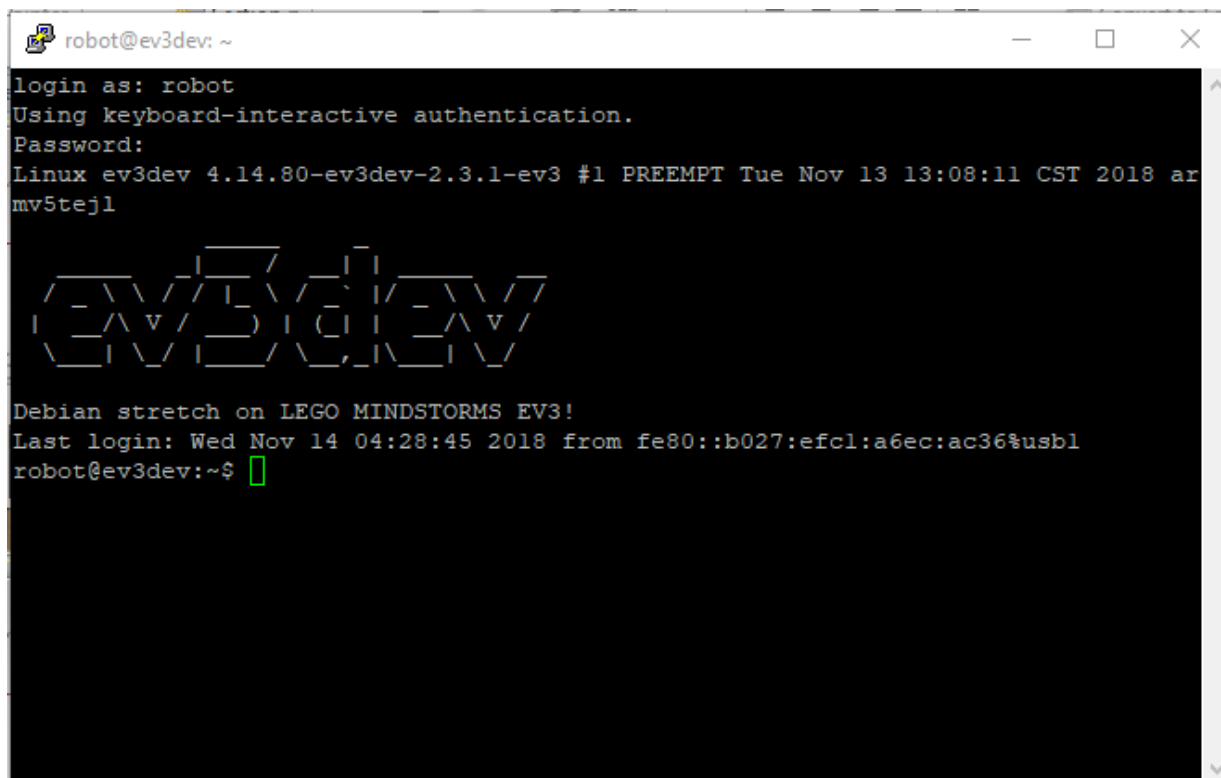
Με την εκτέλεση της εφαρμογής PUTTY, θα σας εμφανιστεί το παρακάτω παράθυρο διαλόγου. Αρχικά πρέπει να γνωρίζετε το όνομα ή το IP του απομακρυσμένου συστήματος, το οποίο πληκτρολογείτε στο πεδίο «**server name**». Προσέξτε η επιλογή «**Port**» να έχει την τιμή «**22**» και η «**Protocol**» να έχει επιλεγμένο το «**SSH**»

Στην δική μας περίπτωση, που θέλουμε να συνδεθούμε με το ρομπότ μας το οποίο βρίσκεται σε λειτουργικό σύστημα ev3dev, πληκτρολογούμε ως Host Name:ev3dev. Στη συνέχεια κάνουμε login χρησιμοποιώντας ως όνομα τη λέξη: robot και ως κωδικό τη λέξη : maker.



Εικόνα 14 Σύνδεση ev3dev στο Putty

Στο παράθυρο που εμφανίζεται κάνουμε login χρησιμοποιώντας ως όνομα τη λέξη: robot και ως κωδικό τη λέξη : maker. Για τις εντολές που βλέπετε θα μιλήσουμε αναλυτικά στο ακριβώς επόμενο κεφάλαιο.



```
robot@ev3dev: ~
login as: robot
Using keyboard-interactive authentication.
Password:
Linux ev3dev 4.14.80-ev3dev-2.3.1-ev3 #1 PREEMPT Tue Nov 13 13:08:11 CST 2018 ar
mv5tej1

ev3dev

Debian stretch on LEGO MINDSTORMS EV3!
Last login: Wed Nov 14 04:28:45 2018 from fe80::b027:efc1:a6ec:ac36%usb1
robot@ev3dev:~$
```

Εικόνα 15 Login και password για σύνδεση στο ev3dev

3.2.3 WinSCP

Το WinSCP είναι ένα πρόγραμμα μεταφοράς αρχείων για Windows το οποίο βασίζεται στο SSH με σκοπό την ασφαλή μεταφορά τους. Το WinSCP υποστηρίζει τις εξής συνδέσεις: SFTP, FTP και SCP για τη μεταφορά αρχείων. Το πρόγραμμα αυτό χρησιμοποιείται κατά βάση για την μεταφορά αρχείων από έναν τοπικό υπολογιστή σε έναν απομακρυσμένο υπολογιστή.

Χρησιμοποιούμε το WinSCP έτσι ώστε να μεταφέρουμε με ασφάλεια αρχεία σε απομακρυσμένους συνεργάτες. [19]

Οι FTP clients είναι εφαρμογές τις οποίες χρησιμοποιούμε ούτως ώστε να επιτύχουμε FTP συνδέσεις σε οποιοδήποτε server. Υπάρχει μεγάλος αριθμός τέτοιων εφαρμογών γνωστών και

μη όπως για παράδειγμα είναι: clients Filezilla για Windows και Mac και το Cyberduck για Mac.

Κάποιες από τις πιο ευρέως γνωστές εφαρμογές κατά τις οποίες μπορούμε να εκτελέσουμε FTP συνδέσεις είναι οι ακόλουθες:

Για προγράμματα FTP:

- ✓ Cute FTP
- ✓ FileZilla
- ✓ WinSCP
- ✓ Smart FTP
- ✓ WS_FTP
- ✓ CyberDuck

Για HTML editors που χρησιμοποιούν FTP:

- ✓ DreamWeaver
- ✓ Expressions Web
- ✓ Notepad ++
- ✓ Sublime
- ✓ iWeb

Για IDE:

- ✓ Eclipse
- ✓ Netbeans
- ✓ Aptana Studio
- ✓ PHP Storm
- ✓ PHP Designer

Οι τρόποι με τους οποίους μπορεί κάποιος να πραγματοποιήσει σύνδεση χρησιμοποιώντας τις παραπάνω εφαρμογές ποικίλει. Μπορούμε να πραγματοποιήσουμε χρήση της εφαρμογής WinSCP εφόσον και αν θέλουμε να κάνουμε συνδέσεις FTPS. Είναι σημαντικό όμως να έχουμε και στοιχεία σύνδεσης επιπέδου root στον server που θα επιδιώξουμε να συνδεθούμε. Ως επί των πλείστων τέτοια δικαιώματα έχουμε σε dedicated ή vps servers.

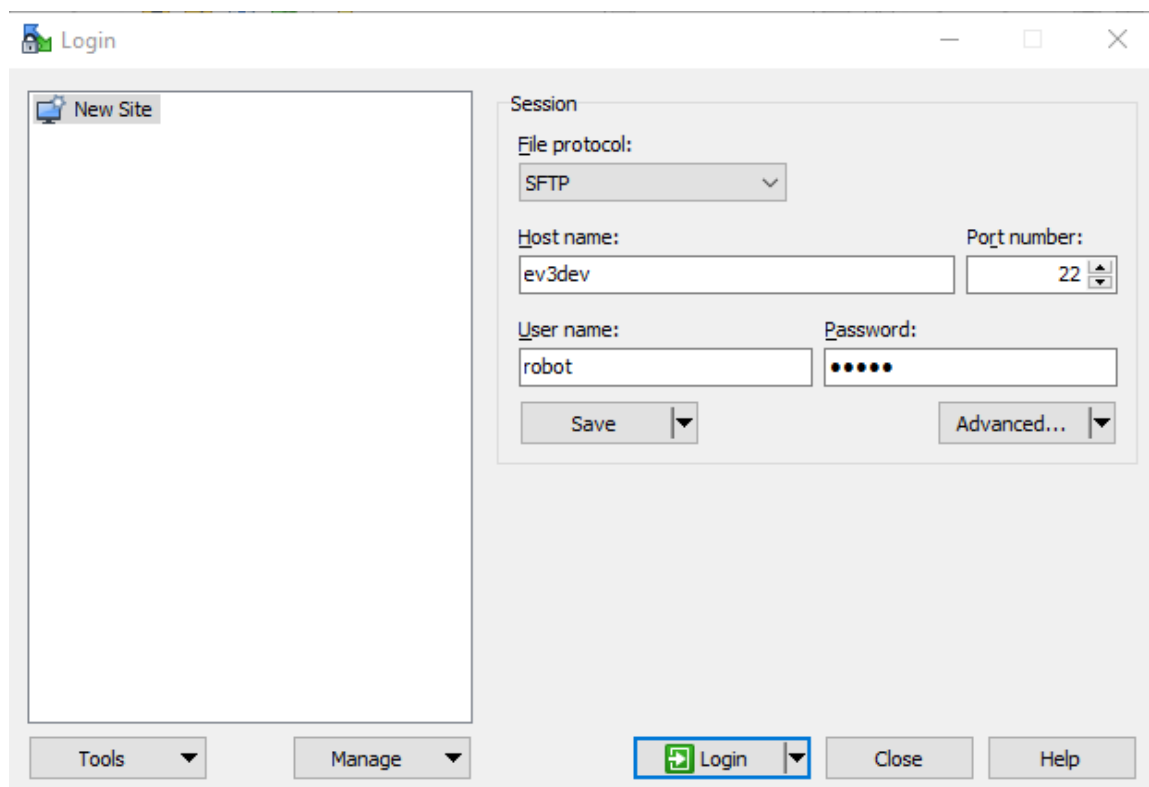
Για τις απλές FTP συνδέσεις, μπορούμε να επιλέξουμε τη χρήση ή μη πιστοποιητικού ασφαλείας κατά τη σύνδεσή μας. Μπορούμε ακόμα να πραγματοποιήσουμε passive mode ή active mode καθώς ακόμα και να προχωρήσουμε σε πιο ειδικευμένες ρυθμίσεις όπως για παράδειγμα τον τύπο της μεταφοράς ASCII Binary ή να ελαχιστοποιήσουμε την ταχύτητα κατά την οποία ολοκληρώνεται το download ή upload των αρχείων μας. [20]

Στην δική μας περίπτωση ανοίγουμε το πρόγραμμα WinSCP και κάνουμε login.

Host name: ev3dev

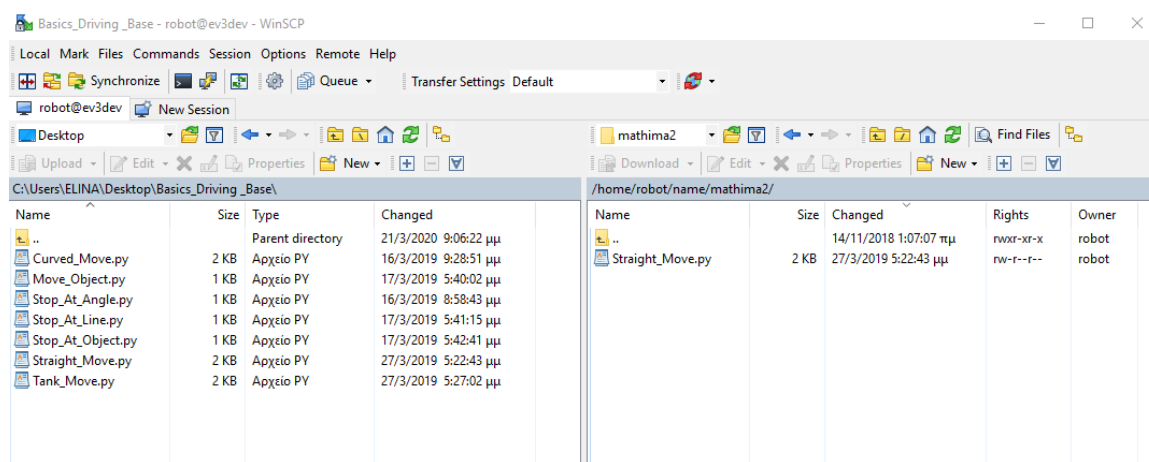
User name: robot

Password: maker



Εικόνα 16 Σύνδεση ev3dev στο WinSCP

Βρίσκουμε από το παράθυρο αριστερά το αποθηκευμένο μας αρχείο σε Python και το σέρνουμε στο δεξιά παράθυρο αφού πρώτα έχουμε εισχωρήσει για παράδειγμα στον φάκελο name και τον υποφάκελο mathima2 όπως δείχνει η παρακάτω εικόνα.



Εικόνα 17 Περιβάλλον WinSCP

3.2.4 Βασικές εντολές πλοήγησης σε λειτουργικό Linux

Ένας υπολογιστής κάνει μόνο δύο πράγματα:

- ✓ Υπολογισμούς
- ✓ Αποθήκευση πληροφοριών

Στο υλικό του υπολογιστή μας (hardware) οι συσκευές με τις οποίες γίνεται η αποθήκευση και η επεξεργασία των πληροφοριών μας είναι:

- ✓ Συσκευή εισόδου
- ✓ Κεντρική μνήμη
- ✓ Κεντρική μονάδα επεξεργασίας (CPU)
- ✓ Συσκευή εξόδου

Γραμμή εντολών:

Η επικοινωνία του χρήστη με τον υπολογιστή γίνεται με τη βοήθεια του λειτουργικού συστήματος. Θα δούμε μία σειρά εντολών (λέξεων) του λειτουργικού συστήματος linux, με τις οποίες μπορούμε να επικοινωνούμε με τον υπολογιστή. Μπορούμε να τις δοκιμάσουμε ανοίγοντας ένα terminal ή το Putty. [21]

Περιήγηση/εκτύπωση/επεξεργασία		Σύστημα	
Cat	Εκτύπωση κειμένου αρχείου	free	Πληροφορίες ελεύθερης μνήμης
Cd	Περιήγηση στους φακέλους	kill	Τερματισμός διεργασιών
Cp	Αντιγραφή αρχείων	reboot	Επανεκκίνηση συστήματος
Echo	Εκτύπωση κειμένου από πηγή	shutdown	Απενεργοποίηση συστήματος
File	Προβολή τύπου αρχείου	startx	Εκκίνηση περιβάλλοντος γραφικών
Head	Προβολή πρώτων γραμμών αρχείου	top	Εκτύπωση τρεχόντων διεργασιών
Tail	Προβολή τελευταίων γραμμών αρχείου	uname	Εκτύπωση διαφόρων

			στοιχείων περί συστήματος
less and more	Προβολή περιεχομένων αρχείου κειμένου	Αναζήτηση	
Ln	Δημιουργία links	find	Αναζήτηση αρχείων
Ls	Προβολή αρχείων ενός φακέλου	locate	Αναζήτηση αρχείων
Man	Προβολή εγχειριδίου βοήθειας εντολής	grep	Εκτύπωση γραμμών κειμένου με τιμή παραμέτρου
Mkdir	Δημιουργία φακέλου	sort	Αναζήτηση εργασιών ταξινόμησης
Mv	Μεταφορά/μετονομασία φακέλου	which	Επιστροφή πλήρης διαδρομής κάποιας εντολής
Nano	Απλός text editor	apropos	Αναζήτηση καταλλήλων εντολών
Pwd	Εκτύπωση ονόματος τρέχοντος φακέλου	Δίκτυα	
Rm	Διαγραφή αρχείων	ifconfig	Ip και ρύθμιση interfaces
Rmdir	Διαγραφή κενού φακέλου	ping	Έλεγχος σύνδεσης με απομακρυσμένο υπολογιστή
Touch	Δημιουργία αρχείου	netstat	Πληροφορίες δικτύων
Tar	Συμπίεση/αποσυμπίεση αρχείων	ssh	Απομακρυσμένη σύνδεση στο τερματικό υπολογιστή
Vim	Εξειδικευμένος text editor		
Wc	Υπολογισμός αριθμού λέξεων/γραμμών/χαρακτήρων ενός κειμένου		

Το τερματικό ενώ είναι ένα πολύ ισχυρό εργαλείο μπορεί να αποβεί και μοιραίο. Αυτό διότι οποιοδήποτε τερματικό λειτουργικού συστήματος μπορεί πολύ εύκολα να καταστρέψει το σύστημά μας. Ακολουθούν κάποιες εντολές οι οποίες θέλουν προσοχή. Παρόλα αυτά οι περισσότερες εντολές απαιτούν να δοθούν με δικαιώματα υπερχρήστη.

Εντολή	Αποτέλεσμα	Αναστρέψιμο	Σχόλια
rm -rf /	Διαγράφει όλα τα	ΟΧΙ	Απαιτεί δικαιώματα

	αρχεία του συστήματος		root
<code>rm -rf /boot/</code>	Διαγράφει τον boot loader του συστήματος	ΝΑΙ	Απαιτεί δικαιώματα root
<code>rm /boot/vmlinuz</code> <code>rm /boot/vmlinuz</code> <code>rm /boot/vmlinuz*</code> <code>rm /boot/vmlinuz*</code>	Διαγράφει τον linux kernel	ΝΑΙ	Απαιτεί δικαιώματα root
<code>mv ~/* /dev/null</code> <code>mv /home/\$USER/* /dev/null</code>	Διαγράφει τα home folder	ΟΧΙ	Δεν απαιτεί δικαιώματα root
<code>mkfs.ext3 /dev/sda</code>	Format σκληρού δίσκου	ΟΧΙ	Απαιτεί δικαιώματα root
<code>any_command > /dev/sda</code>	Καταστρέφει τα δεδομένα του σκληρού δίσκου	ΟΧΙ	Απαιτεί δικαιώματα root
<code>dd if=/dev/urandom of=/dev/sda</code>	Συμπληρώνει τον κενό χώρο του partition σκληρού με δεδομένα	ΟΧΙ	Απαιτεί δικαιώματα root
<code>chmod -R 777 /</code>	Παραχωρεί δικαιώματα root σε όλους τους χρήστες	ΟΧΙ	Απαιτεί δικαιώματα root
<code>chmod 000 -R /</code> <code>chown nobody:nobody -R /</code>	Αφαιρεί πρόσβαση από όλους τους χρήστες εκτός του root	ΟΧΙ	Απαιτεί δικαιώματα root

3.2.5 Εντολές πλοήγησης στο Putty και σύνδεση με εν3dev

Έχουμε κάνει login στο Putty χρησιμοποιώντας ως όνομα τη λέξη: robot και ως κωδικό τη λέξη : maker. Πατώντας ls βλέπουμε ότι δεν υπάρχουν αρχεία και φακέλοι. Δημιουργούμε έναν φάκελο με όνομα student. Ύστερα τον σβήνουμε. Στη συνέχεια δημιουργούμε ένα καινούργιο φάκελο με το όνομα μας. Με την εντολή cd μπορούμε να εισχωρήσουμε μέσα στον φάκελο. Δημιουργούμε μέσα σε αυτόν τον φάκελο άλλον έναν που θα λέγεται mathima2. Μέσα σε αυτόν τον φάκελο μπορούμε να δημιουργήσουμε πρόγραμμα σε Python. Όλες αυτές οι κινήσεις φαίνονται στην παρακάτω εικόνα.

3.3 Λογισμικό Visual Studio Code

Το Visual Studio Code είναι ένα ισχυρό πρόγραμμα επεξεργασίας πηγαίου κώδικα που λειτουργεί στον υπολογιστή μας και είναι διαθέσιμο για Windows, macOS και Linux. Περιλαμβάνει ενσωματωμένη υποστήριξη για Javascript, Typescript και Node.js και διαθέτει ένα μεγάλο εύρος επεκτάσεων για γλώσσες (όπως: C++, C, Java, Python, PHP, Go) καθώς και για προγράμματα .NET και Unity. Οι λειτουργίες του περιλαμβάνουν υποστήριξη για εντοπισμό σφαλμάτων, επισήμανση σύνταξης, έξυπνη αναδιαμόρφωση κώδικα και ενσωματωμένο Git. Οι χρήστες είναι σε θέση να αλλάζουν το θέμα, τις συντομεύσεις πληκτρολογίου αλλά και να εγκαταστήσουν επεκτάσεις που προσθέτουν επιπλέον λειτουργικότητα. Το λογισμικό εγκαθίσταται δωρεάν και από πρόσφατη έρευνα προγραμματιστών κατατάχθηκε ως το πιο δημοφιλές εργαλείο περιβάλλοντος προγραμματιστή. [22]

3.3.1 Σύνδεση EV3DEV στο Visual Studio Code

Καταρχάς είναι απαραίτητο να κατεβάσουμε και να αποσυμπιέσουμε το ακόλουθο link:
<https://github.com/ev3dev/vscodehello-python/archive/master.zip>

Στη συνέχεια ανοίγουμε τον φάκελο στο λογισμικό Visual Studio Code.

Θα δείτε μια ειδοποίηση “Αυτός ο χώρος εργασίας έχει συστάσεις επέκτασης”. Πατήστε “εγκατάσταση όλων”. Ενδέχεται να σας ζητηθεί αργότερα να εγκαταστήσετε το pylint. Σε αυτή την περίπτωση κάντε κλικ στην επιλογή install.

Προαιρετικά, αν θέλετε έλεγχο της έκδοσης λογισμικού, μπορείτε επίσης να εγκαταστήσετε το Git. Το VSC ενσωματώνεται καλά με το Git. Ύστερα εγκαταστήστε το LEGO MINDSTORMS EV3 MicroPython.

Έπειτα δημιουργούμε ένα new project και το αποθηκεύουμε. Στη main μπορούμε να γράψουμε σε Python. Μπορούμε να συνδέσουμε το EV3 μας μέσω USB από το παράθυρο EV3 device browser ή χρησιμοποιώντας WiFi ή Bluetooth επιλέγοντας “Δεν βλέπω τη συσκευή μου” και εισάγοντας μία διεύθυνση IP.

Μπορούμε να δημιουργήσουμε αλλά και να μεταφέρουμε στο VSC οποιοδήποτε αρχείο και να το επεξεργαστούμε σε Python. Για την εκτέλεση του προγράμματος επιλέγουμε το εικονίδιο debugger και έπειτα επιλέγουμε download και run. Τέλος τα προγράμματα μπορούν να εκτελεστούν απευθείας στο BrickManager του robot. [23]

ΡΥTHON ΚΑΙ MICROPΥTHON ΣΕ ΠΕΡΙΒΑΛΛΟΝ VSC

4.1 Η γλώσσα προγραμματισμού Python

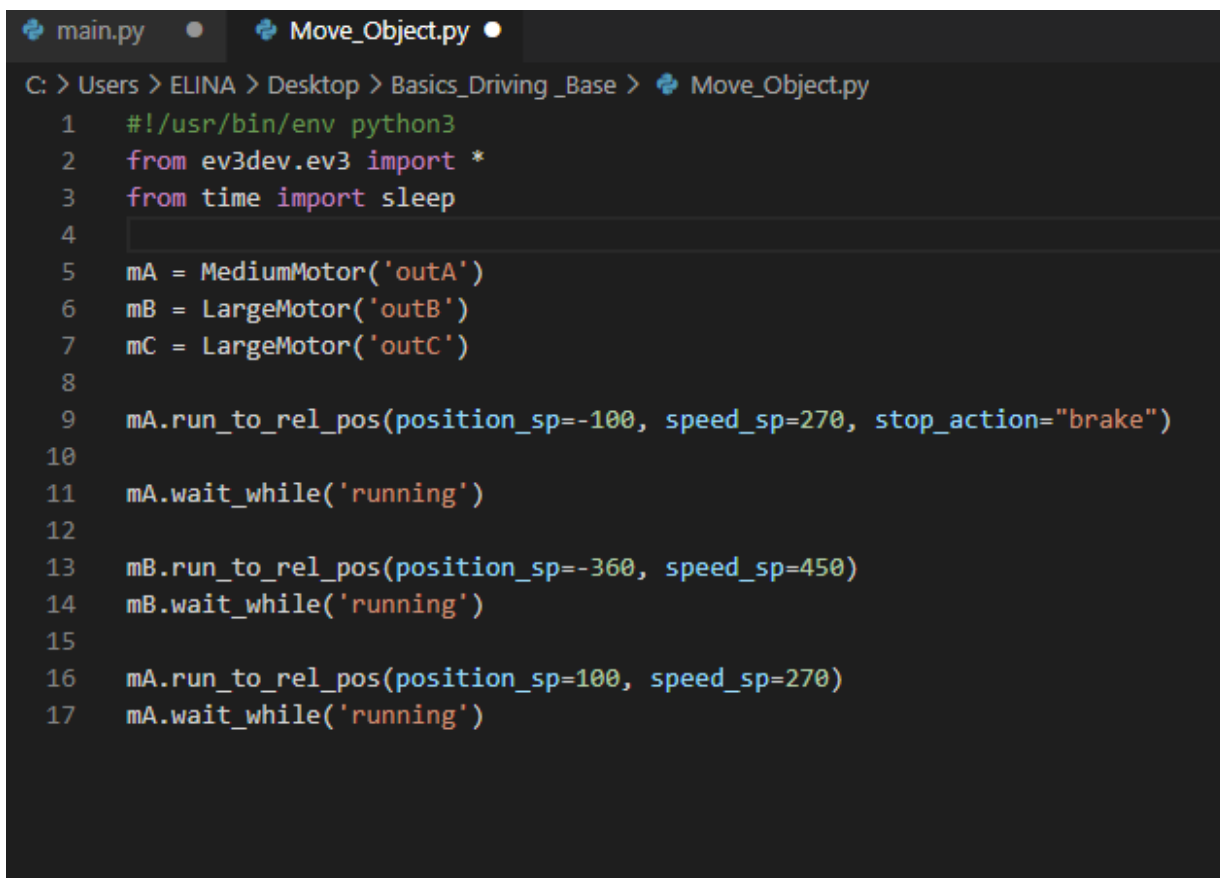
Η python είναι μία ευρέως διαδεδομένη γλώσσα προγραμματισμού υψηλού επιπέδου και γενικού σκοπού. Κυκλοφόρησε για πρώτη φορά το 1991 και η φιλοσοφία σχεδιασμού της δίνει έμφαση στην καλή αναγνωσιμότητα του κώδικα. Οι γλωσσικές δομές της και η αντικειμενοστραφή προσέγγιση στοχεύουν στο να βοηθήσουν στην τεκμηρίωση ενός σαφή και λογικού κώδικα για οποιοδήποτε έργο. Επιπλέον, είναι εφοδιασμένη με πολλές βιβλιοθήκες όπου η κάθε μία από αυτές εκτελεί και τον δικό της σκοπό. Θα αναφέρουμε παρακάτω κάποιες βιβλιοθήκες οι οποίες είναι απαραίτητες για την ολοκλήρωση κώδικα σε περιβάλλον εν3dev. Επιπρόσθετα, είναι σημαντικό να αναφερθεί ότι η Python και γενικότερα ο προγραμματισμός σε διαδεδομένες γλώσσες, μας δίνουν την δυνατότητα να κάνουμε πράγματα που δεν θα ήταν δυνατό με το τυπικό σύστημα προγραμματισμού EV3 όπως για παράδειγμα το να κάνετε το EV3 σας να μιλήσει ότι αγγλικές λέξεις θέλετε. [24]

4.2 Η έκδοση python3 για κατασκευές EV3DEV

Μία βιβλιοθήκη Python3 εφαρμόζει μία διεπαφή για συσκευές εν3dev η οποία μας επιτρέπει να ελέγχουμε κινητήρες, αισθητήρες, κουμπιά υλικού, οθόνες LCD και πολλά άλλα μέσω του κώδικα Python. Απαραίτητη προϋπόθεση είναι να βεβαιωθούμε ότι έχουμε μία έκδοση εν3dev-stretch μεγαλύτερη από 2.2.0. Επιπλέον, μπορούμε να ελέγξουμε την

έκδοση του πυρήνα επιλέγοντας "Πληροφορίες" στο Brickman. Εάν η έκδοση δεν είναι συμβατή πρέπει να την αναβαθμίσουμε.

Κάθε πρόγραμμα Python πρέπει να έχει μερικά βασικά μέρη. Ένα πρότυπο για αρχή είναι το ακόλουθο:



```
main.py ● Move_Object.py ●
C: > Users > ELINA > Desktop > Basics_Driving_Base > Move_Object.py
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  mA = MediumMotor('outA')
6  mB = LargeMotor('outB')
7  mC = LargeMotor('outC')
8
9  mA.run_to_rel_pos(position_sp=-100, speed_sp=270, stop_action="brake")
10
11 mA.wait_while('running')
12
13 mB.run_to_rel_pos(position_sp=-360, speed_sp=450)
14 mB.wait_while('running')
15
16 mA.run_to_rel_pos(position_sp=100, speed_sp=270)
17 mA.wait_while('running')
```

Εικόνα 19 Πρόγραμμα σε python3 από προσωπικό αρχείο εκπαιδευτικών μαθημάτων

Η πρώτη γραμμή είναι απαραίτητο να περιλαμβάνεται σε κάθε πρόγραμμα Python που γράφουμε για ev3dev. Μας επιτρέπει να εκτελούμε αυτό το πρόγραμμα από το Brickman. Οι παρακάτω γραμμές είναι δηλώσεις εισαγωγής που μας δίνουν πρόσβαση στη λειτουργικότητα της βιβλιοθήκης. Σε περίπτωση που θέλουμε να χρησιμοποιήσουμε άλλους τύπους συσκευών ή πρόσθετα βοηθητικά προγράμματα θα πρέπει να προσθέσουμε επιπλέον τάξεις στη λίστα εισαγωγής. Οι γραμμές 5 έως 7 είναι δηλώσεις μεταβλητών mA, mB και mC για χρήση κινητήρα στην θύρα A, B και C αντίστοιχα. Στη γραμμή 9 δηλώνουμε γωνία ίση με -100 και αυτό σημαίνει ότι ο βραχίονας μας θα κινηθεί προς τα κάτω (λόγω της αριστερόστροφης κίνησης του κινητήρα mA). Η γραμμή 11 είναι εντολή επανάληψης ώστε το πρόγραμμα να μην συνεχίσει όσο οι κινητήρες είναι σε λειτουργία. Στη γραμμή 16 παρατηρείται το αντίθετο δηλαδή η γωνία μας είναι ίση με 100 και ο βραχίονας κινείται προς τα πάνω. Ο κινητήρας mB (δηλαδή η αριστερή ρόδα) θα κάνει όπισθεν (γωνία -360) με κίνηση κυκλική εφόσον ο κινητήρας mC δεν κινείται. Η επέκταση .py είναι αυτή που χρησιμοποιούμε για το αρχείο μας. [25]

Παρακάτω αναλύουμε κάποια κομμάτια κώδικα με σκοπό την ανάδειξη επιπρόσθετων επιλογών.

- Έλεγχος LED με τον αισθητήρα αφής

Ο παρακάτω κώδικας ανάβει τις λυχνίες LED κάθε φορά σε κόκκινο όταν πατιέται ο αισθητήρας αφής ενώ σε πράσινο όταν απελευθερώνεται. Μπορούμε να συνδέσουμε τον αισθητήρα αφής σε οποιαδήποτε θύρα πριν το δοκιμάσουμε.

```
ts = TouchSensor()
leds = Leds()

print("Press the touch sensor to change the LED color!")

while True:
    if ts.is_pressed:
        leds.set_color("LEFT", "GREEN")
        leds.set_color("RIGHT", "GREEN")
    else:
        leds.set_color("LEFT", "RED")
        leds.set_color("RIGHT", "RED")
```

Εάν επιθυμούμε να χρησιμοποιήσουμε τον αισθητήρα σε μία συγκεκριμένη θύρα, μπορούμε να καθορίσουμε την θύρα ως εξής:

```
ts = TouchSensor(INPUT_1)
```

- Λειτουργία ενός κινητήρα

Αυτή η επιλογή θα τρέξει έναν κινητήρα στο 75% της μέγιστης ταχύτητας για 5 περιστροφές.

```
m = LargeMotor(OUTPUT_A)
m.on_for_rotations(SpeedPercent(75), 5)
```

Μπορούμε ακόμα να θέσουμε σε λειτουργία έναν κινητήρα για ένα συγκεκριμένο χρονικό διάστημα ή απλά να τον ξεκινήσουμε και να τον αφήσουμε να λειτουργεί μέχρι να του πούμε να σταματήσει.

- Λειτουργία με δύο κινητήρες

Η πιο χαρακτηριστική κίνηση είναι η Tank_Move, η οποία περιλαμβάνεται στα εκπαιδευτικά μαθήματα που έχω συγκεντρώσει. Ο κώδικας αυτός αναφέρεται στην παρακάτω εικόνα.

```
main.py • Move_Object.py • Stop_At_Line.py • Tank_Move_elina.py X
C: > Users > ELINA > Desktop > Basics_Driving_Base > Tank_Move_elina.py
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5
6  mB = LargeMotor('outB') # δήλωση μεταβλητής για χρήση κινητήρα στην θύρα B
7  mC = LargeMotor('outC')
8
9  #δήλωση γωνίας και ταχύτητας κινητήρων
10 mB.run_to_rel_pos(position_sp=685, speed_sp=360, stop_action="brake") #γωνία 68
11 mC.run_to_rel_pos(position_sp=-685, speed_sp=360, stop_action="brake")
12
13
14 #το πρόγραμμα να μην συνεχίσει στις επόμενες εντολές όσο οι κινητήρες είναι σε
15 mB.wait_while('running')
16 mC.wait_while('running')
17
18 #περίμενε 1''
19 sleep(1)
20
21
22 #για όπισθεν -720 μοίρες, το φρένο δεν χρειάζεται να δηλωθεί ξανά
23 mB.run_to_rel_pos(position_sp=1380, speed_sp=360)
24
25
26
27
28 mB.wait_while('running')
29 mC.wait_while('running')
30
31 sleep(1)
32
```

Εικόνα 20 Πρόγραμμα σε python3 από προσωπικό αρχείο εκπαιδευτικών μαθημάτων

Όπως παρατηρούμε σε αυτή την περίπτωση χρησιμεύουν και οι δύο κινητήρες ώστε να πραγματοποιηθεί η πρώτη περιστροφή που εκτελεί το ρομπότ μας.

Γραμμή 10 και 11: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε την ταχύτητα του εκάστοτε κινητήρα και την γωνία. Το ρομπότ για να διαγράψει μία στροφή γύρω από τον εαυτό του θα πρέπει οι δύο κινητήρες να έχουν αντίθετη γωνία η οποία ισούται με 685 μοίρες (μικρός κύκλος), και με ταχύτητα ίδια ώστε να μπορεί να πραγματοποιηθεί μία ολοκληρωμένη στροφή κανονικού κύκλου.

Γραμμή 23: Τώρα το ρομπότ μας για να διαγράψει μία μεγαλύτερη κυκλική πορεία (γύρω από τον εαυτό του) μας αρκεί να είναι σε λειτουργία μόνο ο ένας κινητήρας (κατά προτίμηση ο mB ώστε το ρομπότ να διαγράψει πάλι δεξιόστροφη πορεία).

- Χρήση κειμένου σε ομιλία

Σε τρεις μόνο γραμμές μπορούμε να κάνουμε το ρομπότ μας να μιλήσει χρησιμοποιώντας το `Sound.speak`. Ο τρόπος είναι ο παρακάτω:

```
from ev3dev2.sound import Sound

sound = Sound()
sound.speak('Welcome to the E V 3 dev project!')
```

4.3 Η έκδοση MicroPython

Η MicroPython είναι μία λιτή και αποτελεσματική έκδοση-εφαρμογή της γλώσσας προγραμματισμού Python3.4 που περιλαμβάνει ένα υποσύνολο της τυπικής βιβλιοθήκης Python. Είναι βελτιστοποιημένη για εκτέλεση σε μικροελεγκτές και σε περιορισμένα περιβάλλοντα.

Η EV3 MicroPython λειτουργεί πάνω στο ev3dev με ένα νέο χρόνο εκτέλεσης και βιβλιοθήκη Pybricks MicroPython. Χρησιμοποιείται μέσω της επέκτασης κώδικα του Visual Studio Code και περιλαμβάνει αρκετά πρότυπα έργων για να ξεκινήσει κανείς. Το EV3 MicroPython έχει σχεδιαστεί ώστε να είναι ευκολότερο στη χρήση για αρχάριους, αφαιρώντας κάποια σημεία από την πολυπλοκότητα του προγραμματισμού Python στο EV3.

Επιπρόσθετα, ο χρόνος εκτέλεσης του Pybricks MicroPython (βιβλιοθήκη της MicroPython) προσφέρει ρομπότ EV3 πιο ισχυρά και ευκολότερα στον έλεγχο, προσθέτοντας έτσι πολλές συναρπαστικές δυνατότητες. Αυτό σημαίνει ότι μπορούμε να δημιουργήσουμε πιο σύνθετα προγράμματα και χρήση APIs (υπηρεσίες Internet) για τον ακριβέστερο έλεγχο των κινητήρων. [26]

Όσον αφορά την προετοιμασία μας πριν ξεκινήσουμε να χρησιμοποιούμε MicroPython τα πράγματα είναι πολύ απλά. Κατεβάζουμε αρχικά το αρχείο λήψης EV3 MicroPython micro SD card image. Μπορείτε να το βρείτε στην παρακάτω διεύθυνση: <https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3>

Ύστερα περάστε τον εικονικό δίσκο του MicroPython στην SD κάρτα. Μπορείτε να κατεβάσετε και να εγκαταστήσετε το Etcher: <https://etcher.io>. Εφόσον το εγκαταστήσετε ανοίξτε το πρόγραμμα και επιλέξτε το αρχείο του MicroPython. Εισάγετε την κάρτα μνήμης SD στον υπολογιστή. Επιλέξτε την εγγραφή στην κάρτα μνήμης. Εισάγετε την κάρτα μνήμης στο EV3 και ενεργοποιήστε το.

Αρχικά, θα δείτε την αρχική οθόνη του Mindstorms και το κόκκινο led αναμμένο. Στην συνέχεια αφού φορτώσει θα γίνει πορτοκαλί και θα αναβοσβήνει υποδεικνύοντας την επικοινωνία με την κάρτα SD.

- Σύνδεση EV3Brick με Visual Studio Code
 - 1) Ενεργοποιούμε το EV3 Brick.
 - 2) Συνδέουμε το EV3 Brick με το καλώδιο mini-usb στον υπολογιστή μας.
 - 3) Επιλέγουμε το device browser και αναζητούμε το EV3 Brick μας.
- Η πρώτη και βασική γραμμή πλέον των προγραμμάτων μας αλλάζει:

```
#!/usr/bin/env pybrics-micropython
```

Για να δημιουργήσουμε ένα νέο πρόγραμμα στο Visual Studio Code, ανοίγουμε την καρτέλα EV3 MicroPython και κάνουμε κλικ για να δημιουργήσουμε ένα νέο project. Στη συνέχεια δίνουμε ένα όνομα στο νέο μας project, το αποθηκεύουμε και έπειτα το ανοίγουμε. Το νέο project περιλαμβάνει ήδη ένα αρχείο που ονομάζεται main.py.

Μπορείτε να αναζητήσετε ένα δείγμα γραφής κώδικα σε MicroPython. Ένα δείγμα είναι το παρακάτω.

```
main.py ×
main.py
1  #!/usr/bin/env pybricks-micropython
2  from pybricks.hubs import EV3Brick
3  from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
4  | | | | | | | | | | | | | | | InfraredSensor, UltrasonicSensor, GyroSensor)
5  from pybricks.parameters import Port, Stop, Direction, Button, Color
6  from pybricks.tools import wait, Stopwatch, DataLog
7  from pybricks.robotics import DriveBase
8  from pybricks.media.ev3dev import SoundFile, ImageFile
9
10
11 # This program requires LEGO EV3 MicroPython v2.0 or higher.
12 # Click "Open user guide" on the EV3 extension tab for more information.
13
14
15 # Create your objects here.
16 ev3 = EV3Brick()
17
18
19 # Write your program here.
20 ev3.speaker.beep()
21
```

Εικόνα 21 Δείγμα γραφής κώδικα σε MicroPython

Περιλαμβάνει βιβλιοθήκες και ότι είναι απαραίτητο για να ξεκινήσει κανείς να γράφει ένα απλό πρόγραμμα. Στο σημείο που αναγράφει: Write your program here μπορούμε να συντάξουμε πρόγραμμα.

- Πρόγραμμα ήχου – 1 κινητήρας

Στο σημείο που αναφέρθηκε πριν: Write your program here γράφουμε τις παρακάτω εντολές. Το κομμάτι αυτό κώδικα είναι ικανό να ηχήσει το ρομπότ μας, να περιστρέψει τον κινητήρα και να ηχήσει ξανά με υψηλότερο τόνο.

```
17  brick.sound.beep()
18
19  test_motor = Motor(Port.B)
20
21  test_motor.run_target(500, 90)
22
23  brick.sound.beep(1000,500)
24
25
```

Εικόνα 22 Κομμάτι κώδικα για ήχο

Στη γραμμή 17 παίζει ο πρώτος ήχος.

Στη γραμμή 19 αρχικοποιούμε την θύρα B για έναν κινητήρα.

Στη γραμμή 21 θέτουμε σε λειτουργία τον κινητήρα 500 μοιρών ανά δευτερόλεπτο σε γωνία στόχου 90 μοιρών.

Στη γραμμή 23 παίζει έναν πιο δυνατό ήχο (1000Hz) και με μεγαλύτερη διάρκεια (500ms)

- Αισθητήρας Ultrasonic – 2 κινητήρες

Η παρακάτω κομμάτι κώδικα είναι σε θέση να οδηγήσει το ρομπότ μας μέχρι να βρει κάποιο αντικείμενο.

```
18
19  left = Motor(Port.B)
20  Right = Motor(Port.C)
21  robot = DriveBase(left, right, 56, 114)
22
23  sensor = UltrasonicSensor(Port.S4)
24
25  robot.drive(100, 0)
26  while sensor.distance() > 500:
27      wait(10)
28  robot.stop()
29
```

Εικόνα 23 Κομμάτι κώδικα Αισθητήρα - 2 κινητήρων

Στη γραμμή 19 και 20 αρχικοποιούμε τους δύο μας κινητήρες.

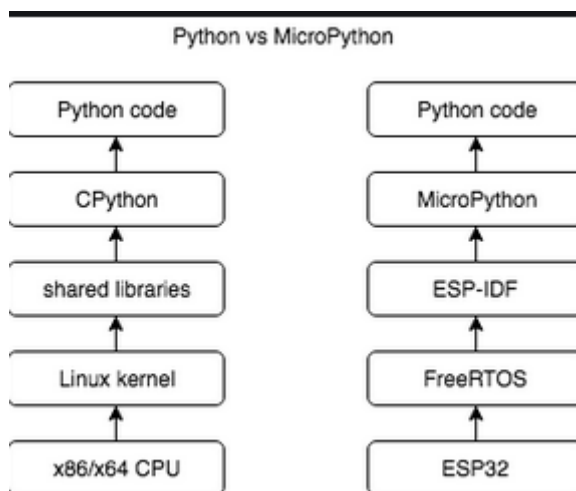
Στη γραμμή 21 αρχικοποιούμε τη βάση κίνησης του ρομπότ μας.

Στη γραμμή 25 έως 28 χρησιμοποιούμε μία δομή επανάληψης ούτως ώστε το ρομπότ μας να κατευθύνεται ευθεία μέχρις ότου εντοπίσει κάποιο αντικείμενο. Μόλις εντοπίσει το αντικείμενο το πρόγραμμα σταματάει να εκτελείται. [27]

4.4 Σύγκριση εκδόσεων python3 και MicroPython

Η MicroPython δεν υποστηρίζει εξ ολοκλήρου τη βασική βιβλιοθήκη Python. Κάποιες εντολές λείπουν κι αυτό διότι δεν υπάρχει δυνατότητα εφαρμογής αυτής της μονάδας για χρήση στον συγκεκριμένο ενσωματωμένο ελεγκτή. Επιπλέον λόγοι είναι η κατανάλωση μνήμης ή η έλλειψη της απαιτούμενης δυνατότητας υλικού.

Η MicroPython δεν μπορεί να εκτελεστεί σε επεξεργαστές που είναι x86. Μία από τις πλατφόρμες που υποστηρίζει η MicroPython είναι το ESP32 (τσιπ). Υλοποιείται στο ESP-IDF το οποίο είναι ένα πλαίσιο που παρέχεται από τον κατασκευαστή του ESP32. Το ESP-IDF βασίζεται σε FreeRTOS πυρήνα λειτουργικού συστήματος σε πραγματικό χρόνο. [28]



Εικόνα 24 Σύγκριση Python3 - MicroPython

Θα θεωρήσουμε ως αναφορά την έκδοση Python3 και με βάση αυτή τη γλώσσα θα συγκρίνουμε την MicroPython. Οι διαφορές τους είναι πολλές και μπορούν να ταξινομηθούν σε 3 κατηγορίες.

- Διαφορές στον Σχεδιασμό

4) Αρχικά, η MicroPython προορίζεται για περιβάλλοντα όπως είναι ένας μικροελεγκτής οι οποίοι έχουν μικρότερες επιδόσεις κυρίως στη μνήμη από ότι τα συστήματα επιτραπέζιων υπολογιστών στα οποία εκτελείται η Python3. Για τον λόγο αυτόν συνεπάγεται ότι η MicroPython πρέπει να σχεδιαστεί λαμβάνοντας σοβαρά υπόψη το γεγονός ότι υπάρχει περιορισμένοι πόροι βγάζοντας στην άκρη περίσια χαρακτηριστικά που δεν ταιριάζουν.

Η MicroPython δεν χρησιμοποιεί την πλήρη γκάμα βιβλιοθηκών της Python. Εξάλλου, οι περισσότερες λειτουργίες αυτών δεν μπορούν να χρησιμοποιηθούν σε μικροεπεξεργαστές και ενσωματωμένα συστήματα, καθώς δεν επαρκεί η μνήμη τέτοιων μικρών συσκευών για την φόρτωση και χρήση των βιβλιοθηκών.

Τελικά, η MicroPython ακολουθεί μία μινιμαλιστική προσέγγιση το οποίο σημαίνει ότι μόνο οι βασικοί τύποι δεδομένων μπορούν να περιληφθούν στον compiler. Όλα τα υπόλοιπα αφήνονται ως εξαρτήσεις τρίτων για συγκεκριμένες εφαρμογές χρηστών. Τέλος, το έργο micropython-lib παρέχει μία μη μονολιθική τυπική βιβλιοθήκη για την MicroPython.

- 5) Η Python3 χρησιμοποιεί την καταμέτρηση αναφοράς (index) ενώ η MicroPython τη συλλογή απορριμάτων (cache) ως κύριο μέσο διαχείρισης της μνήμης.
- 6) Η MicroPython δεν εφαρμόζει πλήρες μοντέλο δεδομένων αντικειμένου αλλά μόνο ένα υποσύνολο αυτού. Η μέθοδος new είναι πιθανό να μην λειτουργεί για προχωρημένες χρήσεις πολλαπλής κληρονομιάς.
- 7) Ο σχεδιασμός MicroPython δεν βασίζεται σε αντικείμενα περιγραφής. Οι περιγραφείς μπορούν να θεωρηθούν ως “υπερδυναμικό χαρακτηριστικό”. Παρόλα αυτά αυτό το χαρακτηριστικό θα μπορούσαμε να πούμε ότι έρχεται σε σύγκρουση με το στόχο να είναι γρήγορο και αποτελεσματικό. Ωστόσο, εφαρμόζεται βασική υποστήριξη για περιγραφείς σε κατηγορίες οι οποίες μπορούν να καθοριστούν από τους χρήστες.
- 8) Το γεγονός ότι μιλάμε για micro σημαίνει ότι υλοποιεί μόνο υποσύνολο λειτουργιών. Οποιοδήποτε ζήτημα μπορεί να αντιμετωπιστεί ως εφαρμογή ώστε να επιλυθεί, χωρίς όμως να σημαίνει ότι θα υπάρξει ποτέ άριστη κάλυψη όλων των δυνατοτήτων της python3.
- 9) Μία ακόμα πραγματικότητα όταν μιλάμε για micro είναι ότι μπορεί να υποστηρίξει μόνο ελάχιστο υποσύνολο χαρακτηριστικών ενδοσκόπησης και προβλημάτων όπως για παράδειγμα είναι τα ονόματα αντικειμένων.
- 10) Η συνάρτηση print() στην MicroPython δεν ελέγχει για αναδρομικές δομές δεδομένων με τον ίδιο τρόπο όπως η python3. Παρόλα αυτά υπάρχουν έλεγχοι για τη χρήση της στοίβας άρα η εκτύπωση αναδρομικής δομής δεδομένων είναι αδύνατο να οδηγήσει σε σφάλμα εξαιτίας της υπερχειλίσης στοίβας.
- 11) Η MicroPython βελτιστοποιεί τον τοπικό χειρισμό μεταβλητών και δεν παρέχει πληροφορίες ενδοσκόπησης για εκείνους.

- Διαφορές εφαρμογής

Κάποιες δυνατότητες δεν καλύπτουν περιορισμένα συστήματα οπότε η εφαρμογή τους τίθεται δύσκολη και μη αποτελεσματική. Οι δυνατότητες αυτές ονομάζονται “διαφορές εφαρμογής” και κάποιες από αυτές είναι πιθανό να εξελιχθούν σε αντικείμενα μελλοντικής ανάπτυξης. Πολλές από αυτές θα επηρεάσουν την απόδοση και την αποτελεσματικότητα οποιασδήποτε εφαρμογής MicroPython, οπότε η μη εφαρμογή κάποιων δυνατοτήτων αναγνωρίζεται από τους στόχους που έχουν τεθεί από την MicroPython.

- 1) Η υποστήριξη Unicode βρίσκεται σε εξέλιξη. Βασίζεται σε εσωτερική αναπαράσταση χρησιμοποιώντας το UTF-8.
- 2) Η οριστικοποίηση αντικειμένου (`_del_`) υποστηρίζεται για ενσωματωμένους τύπους αλλά ακόμα όχι για κλάσεις χρηστών.
- 3) Υπάρχουν κάποιες διαφορές και υποθέσεις συμβατότητας όσον αφορά την κατηγορία των ενσωματωμένων τύπων η οποία εφαρμόζεται ως ένα βαθμό.
- 4) Οι ροές buffer (`io.TextIOWrapper` and `superclasses`) δεν υποστηρίζονται.
- 5) Η λέξη κλειδί `async def` να μην εφαρμόζεται αλλά με κάποιες απλουστεύσεις. Δεν υπάρχει διαφορετικός τύπος αντικειμένου “coroutine” καθώς η λέξη κλειδί ορίζει απλά μία λειτουργία.
- 6) Το `async with` θα πρέπει να είναι διαφορετικό.
- 7) Το `async for` θα πρέπει να είναι διαφορετικό.
- 8) Δεν υπάρχει υποστήριξη για “Μελλοντικά αντικείμενα” με τη μέθοδο `_await_`.
- 9) Η υποστήριξη `Instance_dict_` είναι προαιρετική (απενεργοποιημένη σε πολλές θύρες) και μόνο για ανάγνωση.

- Γνωστά Προβλήματα

Τα γνωστά ζητήματα είναι στην πραγματικότητα σφάλματα ή αλλιώς εσφαλμένες λειτουργίες και παραλείψεις τα οποία βρίσκονται υπό έρευνα και είναι προγραμματισμένο να διορθωθούν. Αυτό σημαίνει ότι η MicroPython δεν παρέχει πλήρη τυπική βιβλιοθήκη. Τα γνωστά αυτά ζητήματα είναι ήδη συγκεντρωμένα και ανοιχτά προς μελέτη από οποιονδήποτε προγραμματιστή. [29]

4.5 Μελλοντικές επεκτάσεις

Οι μελλοντικές επεκτάσεις στον τομέα του προγραμματισμού θα αφορούσαν τη βελτιστοποίηση των γλωσσών προγραμματισμού. Εάν οι γλώσσες προγραμματισμού επόμενης γενιάς περιείχαν προσαρμοστικές μεθόδους και δυνατότητες αυτοεπισκευής, οι οποίες θα μπορούσαν να προσαρμοστούν σε οποιοδήποτε περιβάλλον ή πλατφόρμα, δεν θα υπήρχε ανάγκη ανθρώπινης παρέμβασης (εντοπισμός και ανάκτηση σφαλμάτων). Οι επισκευές θα μπορούσαν να εκτελούνται αυτόματα και αυτό θα άνοιγε ένα νέο επίπεδο γλώσσας προγραμματισμού. Ο προγραμματιστής θα αφιέρωνε περισσότερο χρόνο στην ανάπτυξη παρά στον εντοπισμό σφαλμάτων και την επιδιόρθωση μικρών προβλημάτων κωδικοποίησης σε τεράστιες γραμμές κώδικα. Τα προγράμματα ακόμα καλό θα ήταν να μας επιτρέπουν γραφή κώδικα και για επεξεργαστές υπολογιστών αλλά και για μικροεπεξεργαστές.

Όσον αφορά τον τομέα γενικότερα της εκπαιδευτικής ρομποτικής σαν μάθημα στην τάξη σίγουρα κάποιες βελτιστοποιήσεις στα γραφικά των εγκεφάλων θα ήταν βοηθητικές. Το γραφικό περιβάλλον της οθόνης των εγκεφάλων θα μπορούσε να διαθέτει τα εξής χαρακτηριστικά: μεγαλύτερη φωτεινότητα οθόνης, πιο γρήγορη φόρτωση στην εκκίνηση, καλύτερη ποιότητα μπαταρίας. Ακόμα, γρήγορη συνδεσιμότητα. Θα μπορούσαμε να συνδέουμε μία και μοναδική φορά τον εγκέφαλο με τον υπολογιστή και από την επόμενη φορά να γινότανε αυτόματα. Τα προγράμματα από την άλλη να αναγνωρίζουν απευθείας τις ήδη καταχωρημένες συσκευές που έχουν πραγματοποιήσει σύνδεση από τον υπολογιστή.

Επιπλέον, πολύ χρήσιμο θα ήταν οι εγκέφαλοι να ήταν ταυτόχρονα και μικρότεροι στο μέγεθος καθώς πολλές φορές εμποδίζουν κατά την κατασκευή ενός πολύπλοκου ρομπότ. Το ίδιο θα μπορούσε να ισχύει και για τους αισθητήρες ή ακόμα και για τους κινητήρες. Πρόκειται για αλλαγές τις οποίες εύκολα μπορεί να υποστηρίξει η τεχνολογία την σήμερον ημέρα. Βέβαια η εκπαιδευτική ρομποτική πρόσφατα μπήκε στις τάξεις των σχολείων μας καθώς τα τελευταία μόλις χρόνια έχει γίνει ευρέως γνωστή. Για τον λόγο αυτό είναι πολύ πιθανό γρήγορα να εντυπωσιάσουν καινούργια εκπαιδευτικά πακέτα ρομποτικής τα οποία θα είναι όσο εξελιγμένα τους αρμόζει σε σχέση με την εποχή που ζούμε.

4.6 Συμπεράσματα

Ας ξεκινήσουμε συγκρίνοντας τις δύο παραπάνω εκδόσεις Python τις οποίες και αναλύσαμε. Η Python ανήκει στις δημοφιλέστερες γλώσσες προγραμματισμού και είναι ιδανική γλώσσα για τους προγραμματιστές που κάνουν τα πρώτα τους βήματα. Η φιλική και ευέλικτη σύνταξή του σε συνδυασμό με το γεγονός ότι ακολουθεί μία σειρά ενεργειών, την καθιστά μία ιδιαίτερα καλή επιλογή για τους αρχάριους. Παρόλα αυτά είναι σε θέση να ανταγωνιστεί τις δημοφιλέστερες γλώσσες προγραμματισμού όπως είναι η C και η Java.

Έχουμε λοιπόν αναφερθεί στην Python3 και την MicroPython. Όπως έχουμε ήδη εξηγήσει η MicroPython υστερεί σε κάποια σημεία σε σχέση με την python3. Αυτό συμβαίνει καθώς η MicroPython περιλαμβάνει ένα μικρό υποσύνολο της τυπικής βιβλιοθήκης Python αφού είναι βελτιστοποιημένη για εκτέλεση σε μικροελεγκτές αλλά και σε περιορισμένα περιβάλλοντα. Αυτό όμως δεν σημαίνει ότι μπορούμε να πούμε πως η Python3 είναι μία πιο καλή έκδοση γλώσσας προγραμματισμού. Η κάθε μία εξυπηρετεί τους δικούς της λόγους ύπαρξης και εφαρμόζεται εκεί που μπορεί να προσφέρει περισσότερα.

Η MicroPython είναι σχεδιασμένη για να λειτουργεί σε μικροελεγκτές και σε περιορισμένα περιβάλλοντα. Το MicroPython pyboard είναι μια συμπαγής ηλεκτρονική πλακέτα κυκλώματος που τρέχει MicroPython στο γυμνό μέταλλο, δίνοντάς σας ένα λειτουργικό σύστημα Python χαμηλού επιπέδου που μπορεί να χρησιμοποιηθεί για τον έλεγχο όλων των ειδών ηλεκτρονικών έργων.

Το MicroPython είναι γεμάτο προηγμένες δυνατότητες, όπως μια διαδραστική προτροπή, αυθαίρετοι ακέραιοι ακρίβεια, κλείσιμο, κατανόηση λίστας, χειρισμός εξαιρέσεων και άλλα. Ωστόσο, είναι αρκετά συμπαγής ώστε να χωράει και να λειτουργεί μέσα σε μόλις 256k χώρο κώδικα και 16k RAM.

Όσον αφορά όμως τις δυνατότητες που έχει η κάθε γλώσσα για να υποστηρίξει και να εξελίξει τις ρομποτικές κινήσεις αλλά και την τεχνητή νοημοσύνη ενός EV3 ρομπότ τότε μάλλον γνωρίζουμε ποια θα ήταν η καταλληλότερη έκδοση. Η python3 μας προσφέρει ένα μεγάλο εύρος βιβλιοθηκών ικανό να μας προσφέρει πολλές επιλογές. Εξάλλου, η MicroPython είναι μία έκδοση που προσπαθεί να φτάσει το επίπεδο βιβλιοθηκών της python.

Από την άλλη όμως, η MicroPython έχει να προσφέρει τρομερές προγραμματιστικές γνώσεις αρκεί μόνο να αρχίσει κανείς να την μαθαίνει αφού πρώτα έχει φτάσει σε καλό επίπεδο γνώσεων της python.

Πιο γενικά λοιπόν, η εκπαιδευτική ρομποτική, με τη βοήθεια του πακέτου Lego Mindstorms EV3, δίνουν την δυνατότητα στα παιδιά να αποκομίσουν ένα τεράστιο εύρος γνώσεων. Η μάθηση μέσω του συγκεκριμένου πακέτου δεν σταματάει ποτέ καθώς οι γλώσσες προγραμματισμού που μπορεί να υποστηρίξει είναι πολλές. Εκτός από την Python άλλες διαθέσιμες γλώσσες είναι: Java, Go, C++, C, Prolog, Vala, Genie και άλλες γλώσσες βασισμένες στο GObject με ev3devKit. Ακόμα ορισμένες από τις παραπάνω βιβλιοθήκες εφαρμόζουν διεπαφές και για άλλες γλώσσες όπως: Ruby και Perl με ev3dev-c. Όπως εύκολα

καταλαβαίνουμε τα εφόδια που μπορεί να μας δώσει η εκπαιδευτική ρομποτική φτάνουν σε επίπεδα ακόμα και επαγγελματικής αποκατάστασης.

Εκτός από τον προγραμματισμό που στις μέρες μας ειδικά είναι μεγάλο προσόν να κατέχει κανείς, η εκπαιδευτική ρομποτική μπορεί ακόμα να προσφέρει πολλά και στο μηχανολογικό κομμάτι μέσω των κατασκευών. Τα projects που είναι ανοιχτά προς όλους γενικά στο Internet αλλά και στην επίσημη σελίδα en3dev.drg, είναι αμέτρητα και σε όλες τις βαθμίδες δυσκολίας. Το να ακολουθήσει κανείς τις έτοιμες οδηγίες για την κατασκευή ενός ρομπότ είναι το μόνο εύκολο. Μετά από αυτό το στάδιο όμως ο μαθητής θα νιώσει μεγάλη ικανοποίηση δημιουργώντας μεγάλες κατασκευές και η επόμενη κίνηση θα είναι η υλοποίηση δικού του project. Αυτό θα τον κάνει να ασχοληθεί με το μηχανολογικό κατασκευαστικό κομμάτι, τα μαθηματικά ακόμα και τη φυσική.

Συνοψίζοντας λοιπόν, η εκπαιδευτική ρομποτική αποτελεί στις μέρες μας το πιο δημοφιλή, καινοτόμο και εξελίξιμο μάθημα που έχει ποτέ εισαχθεί στα σχολεία. Το μόνο που αρκεί είναι όλα τα παιδιά να γνωρίσουν αυτό το αντικείμενο και όσοι το αγαπήσουν με όρεξη για δουλεία, περιέργεια και υπομονή μπορούν να κερδίσουν τον κόσμο.

ΠΑΡΑΡΤΗΜΑ

Εκπαιδευτικά μαθήματα σε python3 και Putty

[i] Κώδικας Ευθεία Κίνηση – Straight Move

```
Straight_Move.py x
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  mB = LargeMotor('outB')
6  mC = LargeMotor('outC')
7
8  mB.run_to_rel_pos(position_sp=720, speed_sp=450, stop_action="brake")
9  mC.run_to_rel_pos(position_sp=720, speed_sp=450, stop_action="brake")
10
11 mB.wait_while('running')
12 mC.wait_while('running')
13
14 sleep(1)
15
16 mB.run_to_rel_pos(position_sp=-720, speed_sp=450)
17 mC.run_to_rel_pos(position_sp=-720, speed_sp=450)
18
19 mB.wait_while('running')
20 mC.wait_while('running')
21
22 sleep(1)
23
24 mB.run_timed(time_sp=1000, speed_sp=450)
25 mC.run_timed(time_sp=1000, speed_sp=450)
26
27 mB.wait_while('running')
28 mC.wait_while('running')
```

Επεξήγηση προγράμματος:

Γραμμή 1: Δήλωση αρχείου τύπου python 3 script

Γραμμή 2 και 3: Δήλωση βιβλιοθηκών όπου μέσα σε αυτές βρίσκονται οι εντολές που θα χρησιμοποιήσουμε.

Γραμμή 5 και 6: Δήλωση μεταβλητών mB και mC για χρήση κινητήρα στην θύρα B και C αντίστοιχα.

Γραμμή 8 και 9: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε την ταχύτητα του εκάστοτε κινητήρα όπου μέγιστη ταχύτητα $\max=900$ και την γωνία ($=720$) όπου πραγματοποιεί την πορεία ευθεία μπροστά.

Γραμμή 11 και 12: Εντολές επανάληψης ώστε το πρόγραμμα να μην συνεχίσει όσο οι κινητήρες είναι σε λειτουργία.

Γραμμή 14: Εντολή παύσης ενός δευτερολέπτου.

Μέχρι εδώ το ρομπότ μας έχει πραγματοποιήσει την κίνησή του προς τα εμπρός. Πάμε να συνεχίσουμε για την ίδια ακριβώς κίνηση προς τα πίσω.

Γραμμή 16: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε την γωνία (πλέον -720 για κίνηση προς τα πίσω) και την ταχύτητα των κινητήρων. Στην όπισθεν το φρένο δεν χρειάζεται να δηλωθεί ξανά.

Γραμμή 19 και 20: Εντολές επανάληψης ώστε το πρόγραμμα να μην συνεχίσει όσο οι κινητήρες είναι σε λειτουργία.

Γραμμή 22: Εντολή παύσης ενός δευτερολέπτου.

Μέχρι εδώ το ρομπότ μας έχει πραγματοποιήσει την κίνηση όπισθεν. Έχει ξεκινήσει δηλαδή από όπου ξεκίνησε.

Γραμμή 24 και 25: Ίδια κίνηση εμπρός με συνάρτηση χρόνου=1000 και ταχύτητα 450.

Γραμμή 27 και 28: Εντολές επανάληψης ώστε το πρόγραμμα να μην συνεχίσει όσο οι κινητήρες είναι σε λειτουργία.

Πλέον έχει πραγματοποιηθεί μία μικρή κίνηση προς τα εμπρός η οποία αντιστοιχεί σε χρόνο=1000.

[ii] Κώδικας Περιστροφή – Move Tank

```
Tank_Move_elina.py X
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  mB = LargeMotor('outB')
6  mC = LargeMotor('outC')
7
8  mB.run_to_rel_pos(position_sp=685, speed_sp=360, stop_action="brake")
9  mC.run_to_rel_pos(position_sp=-685, speed_sp=360, stop_action="brake")
10
11  mB.wait_while('running')
12  mC.wait_while('running')
13
14  sleep(1)
15
16  mB.run_to_rel_pos(position_sp=1380, speed_sp=360)
17
18  mB.wait_while('running')
19  mC.wait_while('running')
20
21  sleep(1)
22
23
```

Επεξήγηση προγράμματος:

Γραμμή 8 και 9: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε την ταχύτητα του εκάστοτε κινητήρα και την γωνία. Το ρομπότ για να διαγράψει μία στροφή γύρω από τον εαυτό του θα πρέπει οι δύο κινητήρες να έχουν αντίθετη γωνία η οποία ισούται με 685 μοίρες (μικρός κύκλος), και με ταχύτητα ίδια ώστε να μπορεί να πραγματοποιηθεί μία ολοκληρωμένη στροφή κανονικού κύκλου.

Γραμμή 16: Τώρα το ρομπότ μας για να διαγράψει μία μεγαλύτερη κυκλική πορεία (γύρω από τον εαυτό του) μας αρκεί να είναι σε λειτουργία μόνο ο ένας κινητήρας (κατά προτίμηση ο mB ώστε το ρομπότ να διαγράψει πάλι δεξιόστροφη πορεία).

[iii] Κώδικας Stop σε αντικείμενο – Stop at Object

```
Stop_At_Object.py X
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  us = UltrasonicSensor()
6  us.mode='US-DIST-CM'
7
8  mB = LargeMotor('outB')
9  mC = LargeMotor('outC')
10
11  startdistance = us.value()
12
13  mB.run_forever(speed_sp=450)
14  mC.run_forever(speed_sp=450)
15
16  while us.value() > startdistance-110:
17      sleep(0.01)
18
19  mB.stop(stop_action="brake")
20  mC.stop(stop_action="brake")
21
22  sleep(1)
23
24  mB.run_forever(speed_sp=-450)
25  mC.run_forever(speed_sp=-450)
26
27  while us.value() < startdistance-50:
28      sleep(0.01)
29
30  mB.stop()
31  mC.stop()
```

Επεξήγηση Προγράμματος:

Γραμμή 5 και 6: Δήλωση αισθητήρα του Ultrasonic στην μεταβλητή us και δήλωση απόστασης σε εκατοστά.

Γραμμή 11: Αρχικοποίηση τιμής του αισθητήρα Ultrasonic στην μεταβλητή startdistance.

Γραμμή 13 και 14: Εντολές ενεργοποίησης των κινητήρων σε μία συγκεκριμένη ταχύτητα (ευθύγραμμη κίνηση).

Γραμμή 16 και 17: Δομή επανάληψης η οποία ελέγχει εάν ο αισθητήρας έχει κάποιο εμπόδιο μπροστά του σε απόσταση μικρότερη συγκεκριμένων εκατοστών. Εάν υπάρχει εμπόδιο γίνεται μία μικρή παύση του προγράμματος.

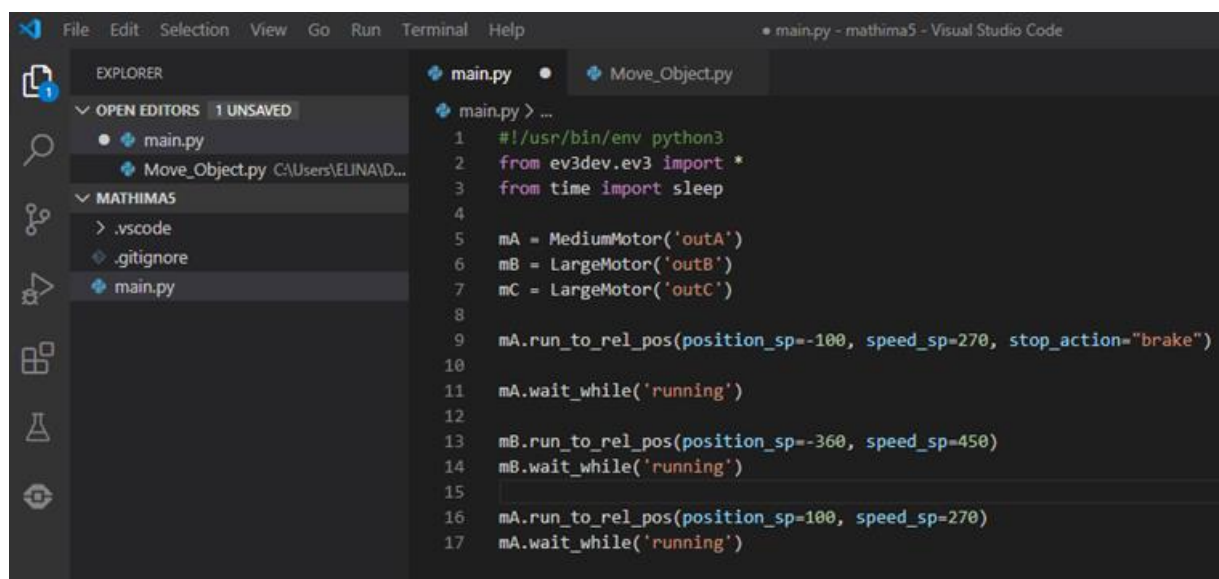
Γραμμή 19 και 20: Ακινητοποίηση των κινητήρων.

Γραμμή 24 και 25: Ενεργοποίηση των κινητήρων με μία αρνητική ταχύτητα (οπισθοδρόμηση).

Γραμμή 27 και 28: Δομή επανάληψης η οποία ελέγχει εάν ο αισθητήρας έχει το εμπόδιο μπροστά του σε απόσταση μεγαλύτερη από κάποια συγκεκριμένα εκατοστά. Εάν επιτευχθεί η επιθυμητή απόσταση γίνεται μία μικρή παύση του προγράμματος.

Εκπαιδευτικά μαθήματα σε python3 και Visual Studio Code

[i] Κώδικας Μετακίνηση Αντικειμένου – Move Object



```
File Edit Selection View Go Run Terminal Help
main.py - mathima5 - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
main.py
Move_Object.py C:\Users\EJUNAID...
MATHIMAS
.vscode
.gitignore
main.py

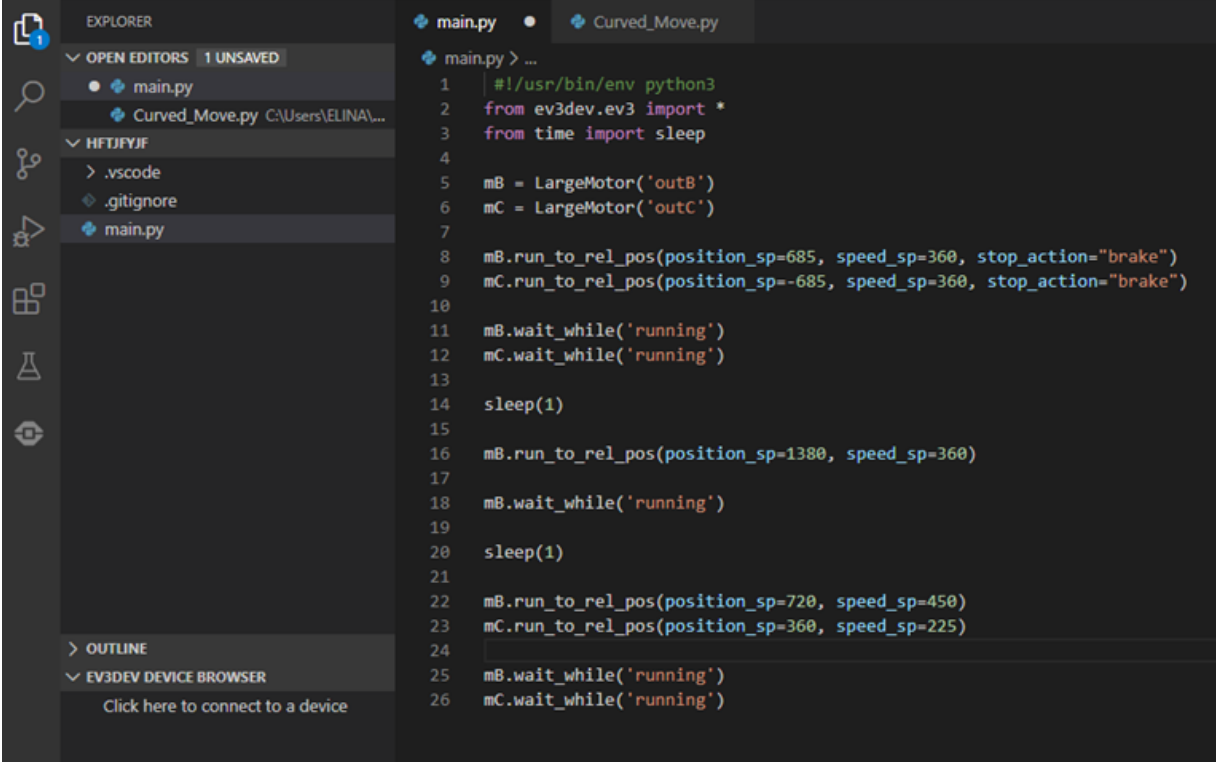
main.py
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  mA = MediumMotor('outA')
6  mB = LargeMotor('outB')
7  mC = LargeMotor('outC')
8
9  mA.run_to_rel_pos(position_sp=-100, speed_sp=270, stop_action="brake")
10
11 mA.wait_while('running')
12
13 mB.run_to_rel_pos(position_sp=-360, speed_sp=450)
14 mB.wait_while('running')
15
16 mA.run_to_rel_pos(position_sp=100, speed_sp=270)
17 mA.wait_while('running')
```

Επεξήγηση Προγράμματος:

Γραμμή 9: δηλώνουμε γωνία ίση με -100 και αυτό σημαίνει ότι ο βραχίονας μας θα κινηθεί προς τα κάτω (λόγω της αριστερόστροφης κίνησης του κινητήρα mA) .

Γραμμή 16: Στην αντίθετη περίπτωση όπου η γωνία μας είναι ίση με 100 ο βραχίονας κινείται προς τα πάνω. Ο κινητήρας mB (δηλαδή η αριστερή ρόδα) θα κάνει όπισθεν (γωνία -360) με κίνηση κυκλική εφόσον ο κινητήρας mC δεν κινείται.

[ii] Κώδικας Καμπυλωτή Κίνηση – Curved Move



```
main.py > ...
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  mB = LargeMotor('outB')
6  mC = LargeMotor('outC')
7
8  mB.run_to_rel_pos(position_sp=685, speed_sp=360, stop_action="brake")
9  mC.run_to_rel_pos(position_sp=-685, speed_sp=360, stop_action="brake")
10
11  mB.wait_while('running')
12  mC.wait_while('running')
13
14  sleep(1)
15
16  mB.run_to_rel_pos(position_sp=1380, speed_sp=360)
17
18  mB.wait_while('running')
19
20  sleep(1)
21
22  mB.run_to_rel_pos(position_sp=720, speed_sp=450)
23  mC.run_to_rel_pos(position_sp=360, speed_sp=225)
24
25  mB.wait_while('running')
26  mC.wait_while('running')
```

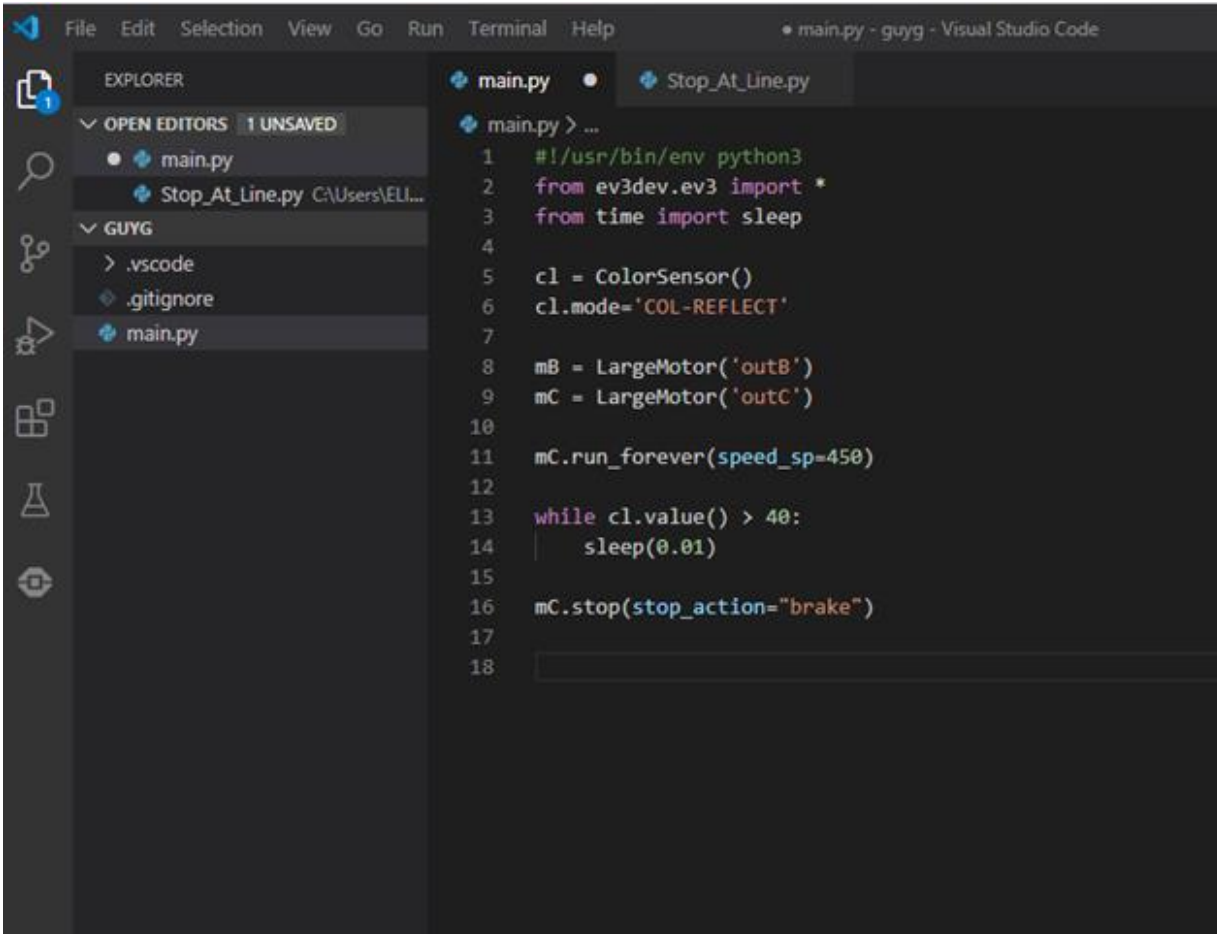
Επεξήγηση προγράμματος:

Γραμμή 8 και 9: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε γωνία και ταχύτητα κινητήρων. Το ρομπότ για να διαγράψει μία στροφή γύρω από τον εαυτό του (τον πιο μικρό κύκλο που θα μπορούσε να εκτελέσει) θα πρέπει οι δύο κινητήρες να έχουν αντίθετη γωνία η οποία ισούται με 685, και με ταχύτητα ίδια ώστε να μπορεί να πραγματοποιηθεί μία ολοκληρωμένη στροφή κανονικού κύκλου.

Γραμμή 16: Τώρα το ρομπότ μας για να διαγράψει μία μεγαλύτερη κυκλική πορεία (γύρω από τον εαυτό του) μας αρκεί να είναι σε λειτουργία μόνο ο ένας κινητήρας (κατά προτίμηση ο mB ώστε το ρομπότ να διαγράψει πάλι δεξιόστροφη πορεία).

Γραμμή 22 και 23: Η τελευταία κίνηση είναι μία κίνηση με δεξιόστροφη φορά η οποία επιτυγχάνεται εφόσον ο κινητήρας mB έχει μεγαλύτερη ταχύτητα από τον mC.

[iii] Κώδικας Stop στη γραμμή – Stop at Line



```
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import *
3  from time import sleep
4
5  cl = ColorSensor()
6  cl.mode='COL-REFLECT'
7
8  mB = LargeMotor('outB')
9  mC = LargeMotor('outC')
10
11  mC.run_forever(speed_sp=450)
12
13  while cl.value() > 40:
14      sleep(0.01)
15
16  mC.stop(stop_action="brake")
17
18
```

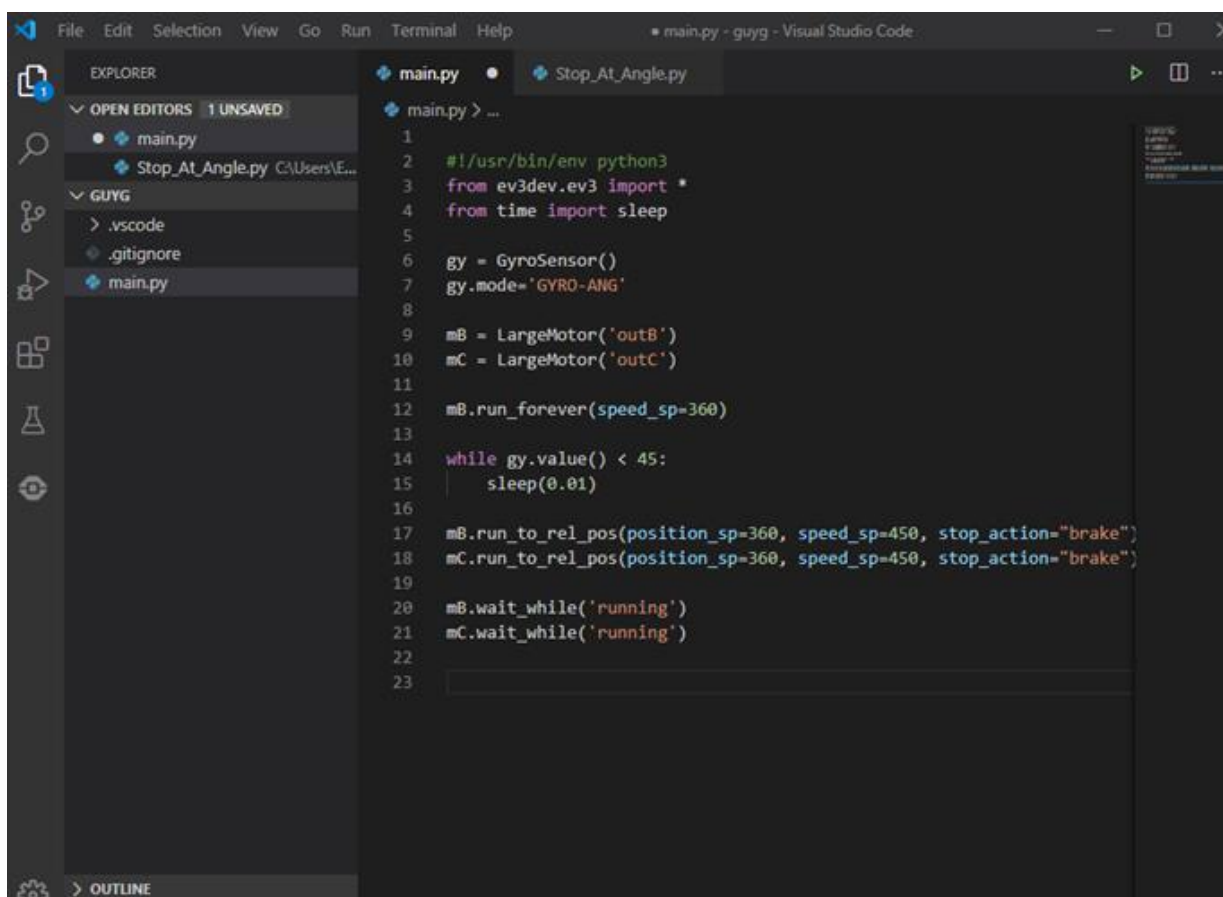
Επεξήγηση Προγράμματος:

Γραμμή 5 και 6: Δήλωση του αισθητήρα ColoSensor στην μεταβλητή c1 και δήλωση c1.mode για τη μέτρηση της έντασης ανακλώμενου φωτός. Σε αυτή τη λειτουργία ο αισθητήρας θα επιστρέψει μία τιμή ανάμεσα στο 0 και το 100.

Γραμμή 11: Εντολή ενεργοποίησης του κινητήρα σε μία συγκεκριμένη ταχύτητα (ευθύγραμμη κίνηση).

Γραμμή 13 και 14: Δομή επανάληψης που ελέγχει πότε ο αισθητήρας χρώματος θα ανιχνεύσει ότι η φωτεινότητα έχει πέσει κάτω από μία συγκεκριμένη τιμή η οποία είναι το 40. (Η επανάληψη αυτή γίνεται συνεχώς όσο το ρομπότ μας κινείται πάνω σε μία λευκή επιφάνεια.) Εάν επιτευχθεί η επιθυμητή ανίχνευση του μαύρου χρώματος, γίνεται μία μικρή παύση του προγράμματος.

[iv] Κώδικας Stop σε συγκεκριμένη γωνία – Stop at Angle



```
1
2 #!/usr/bin/env python3
3 from ev3dev.ev3 import *
4 from time import sleep
5
6 gy = GyroSensor()
7 gy.mode='GYRO-ANG'
8
9 mB = LargeMotor('outB')
10 mC = LargeMotor('outC')
11
12 mB.run_forever(speed_sp=360)
13
14 while gy.value() < 45:
15     sleep(0.01)
16
17 mB.run_to_rel_pos(position_sp=360, speed_sp=450, stop_action="brake")
18 mC.run_to_rel_pos(position_sp=360, speed_sp=450, stop_action="brake")
19
20 mB.wait_while('running')
21 mC.wait_while('running')
22
23
```


Επεξήγηση Προγράμματος:

Γραμμή 6 και 7: Δήλωση του αισθητήρα GyroSensor στην μεταβλητή gy και δήλωση gy.mode για τη μέτρηση της γωνίας στροφής σε μοίρες.

Γραμμή 12: Εντολή ενεργοποίησης του κινητήρα αριστερά με ταχύτητα 40% για να πραγματοποιηθεί μία μεσαία στροφή δεξιά.

Γραμμή 14 και 15: Δομή επανάληψης που ελέγχει πότε ο αισθητήρας θα ανιχνεύσει ότι το ρομπότ έχει γυρίσει (τουλάχιστον) 45 μοίρες προς τη θετική κατεύθυνση δεξιά. Εάν επιτευχθεί η επιθυμητή ανίχνευση των 45 μοιρών γίνεται μία μικρή παύση του προγράμματος.

Γραμμή 17 και 18: Εντολές για τους δύο κινητήρες όπου ως δεδομένα εισόδου έχουμε γωνία και ταχύτητα κινητήρων. Το ρομπότ κινείται ευθεία και οι τροχοί έχουν πραγματοποιήσει μια περιστροφή 360 μοίρες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ρομπότ – Wikipedia. Πηγή:

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

[2] Πολυξένη Κουτσίκου 2012, Εργαστήριο Ρομποτικής. Πηγή:

<http://users.sch.gr/jenyk/index.php/artificialintelligence/ai-historicalreview/11-robotics/17-whatisroboticswhatisrobot>

[3] Βιβλίο "Βασικές αρχές ρομποτικής", Maja J Mataric, Εκδόσεις Κλειδάριθμος. Ποια είναι η προέλευση των ρομπότ; Μια σύντομη, αλλά συναρπαστική ιστορική αναδρομή στη ρομποτική.

[4] Ρομποτική και Τεχνητή Νοημοσύνη - 2ο ΓΕΛ Περιστερίου - Ερευνητική εργασία Β΄ τάξης 2013-2014 – Ημερολόγιο. Πηγή:

<https://projectrobotics.wordpress.com/team1/%CE%BF%CF%81%CE%B9%CF%83%CE%B9%CF%8C%CE%B9-%CF%84%CF%89%CE%BD-%CF%81%CE%BF%CE%BC%CF%80%CF%8C%CF%84/>

[5] Βιβλίο "Εισαγωγή στην επιστήμη των υπολογιστών" Behrouz Forouzan και Firouz Mosharraf, Δεύτερη Έκδοση. Τεχνητή Νοημοσύνη

[6] Ρομποτική και Τεχνητή Νοημοσύνη - 2ο ΓΕΛ Περιστερίου - Ερευνητική εργασία Β΄ τάξης 2013-2014 – Ημερολόγιο. Πηγή:

<https://projectrobotics.wordpress.com/ae/bio/>

[7] Εκπαιδευτική Ρομποτική – Wikipedia. Πηγή:

https://el.wikipedia.org/wiki/%CE%95%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%85%CF%84%CE%B9%CE%BA%CE%AE_%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE

[8] Πολυξένη Κουτσίκου 2012, Εργαστήριο Ρομποτικής. Πηγή:

<http://users.sch.gr/jenyk/index.php/educationalrobotics>

[9] Υδρογείος Education - Επωνυμία εταιρείας Χουρίδου Καρολίνα. Πηγή:

<https://idrogiος.com/h-didaskalia-toy-programmatismoy-se-paidia>

[10] LEGO MINDSTORMS EDUCATION EV3, 2019 The LEGO Group. Πηγή:

<https://education.lego.com/en-us/support/mindstorms-ev3>

[11] Πληροφορίες εκπαιδευτικού EV3 Core Set, LEGO Education MINDSTORMS

EV3, Διερευνητική Μάθηση. Πηγή:

<https://www.why.gr/%CE%BA%CE%B1%CF%84%CE%B1%CF%83%CF%84%CE%B7%CE%BC%CE%B1/%ce%b5%ce%ba%cf%80%ce%b1%ce%b9%ce%b4%ce%b5%cf%85%cf%84%ce%b9%ce%ba%ce%ac-%cf%80%ce%b1%ce%ba%ce%ad%cf%84%ce%b1/stem/stem-%ce%b4%ce%b5%cf%85%cf%84%ce%b5%cf%81%ce%bf%ce%b2%ce%ac%ce%b8%ce%bc%ce%b9%ce%b1%cf%82-%ce%b5%ce%ba%cf%80%ce%b1%ce%af%ce%b4%ce%b5%cf%85%cf%83%ce%b7%cf%82/lego-education-mindstorms-ev3/>

[12] Υδρόγειος Education - Επωνυμία εταιρείας Χουρίδου Καρολίνα. Πηγή:

<https://idrogiος.com/lego-mindstorms-ev3-ena-set-gia-to-spiti-h-to-sxoleio>

[13] Πληροφορίες εκπαιδευτικού EV3 Core Set, LEGO Education MINDSTORMS EV3, Διερευνητική Μάθηση. Πηγή:

<https://www.why.gr/%CE%BA%CE%B1%CF%84%CE%B1%CF%83%CF%84%CE%B7%CE%BC%CE%B1/lego-education/lego-education-mindstorms/lego-education-mindstorms-ev3-ultrasonic-sensor/>

[14] Πληροφορίες εκπαιδευτικού EV3 Core Set, LEGO Education MINDSTORMS EV3, Διερευνητική Μάθηση. Πηγή:

<https://www.why.gr/%CE%BA%CE%B1%CF%84%CE%B1%CF%83%CF%84%CE%B7%CE%BC%CE%B1/lego-education/lego-education-mindstorms/lego-education-mindstorms-ev3-large-servo-motor/>

[15] Βιβλίο "C++ Προγραμματισμός" P.J DEITEL, H.M.DEITEL, Εκδόσεις Μ. Γκιούρδας. Τα πρώτα λειτουργικά συστήματα.

[16] Λειτουργικό Σύστημα – Wikipedia. Πηγή:

https://el.wikipedia.org/wiki/%CE%9B%CE%B5%CE%B9%CF%84%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CF%8C_%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1

[17] EV3DEV.DRG, Getting started with ev3dev. Download the latest ev3dev image file. Πηγή:

<https://www.ev3dev.org/docs/getting-started/>

[18] SSH – Wikipedia. Πηγή:

<https://el.wikipedia.org/wiki/SSH>

[19] Download astro.com, DownloadAstro.com 2020. Μεταφορά αρχείων μεταξύ τοπικών και απομακρυσμένων υπολογιστών. Πηγή:

<https://winscp.el.downloadastro.com/>

[20] ip.gr, Web Hosting and Domain Name registration services in Greece. Τι είναι ο FTP client. Πηγή:

https://www.ip.gr/Hosting/%CE%A4%CE%B9_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9_%CE%BF_FTP_client-216.html

[21] ΒΙΚΙΒΙΒΛΙΑ, ανοιχτά βιβλία για έναν ανοιχτό κόσμο. Linux για αρχάριους / Βασικές γνώσεις τερματικού. Πηγή:

https://el.wikibooks.org/wiki/Linux_%CE%B3%CE%B9%CE%B1_%CE%B1%CF%81%CF%87%CE%AC%CF%81%CE%B9%CE%BF%CF%85%CF%82/%CE%92%CE%B1%CF%83%CE%B9%CE%BA%CE%AD%CF%82_%CE%B3%CE%BD%CF%8E%CF%83%CE%B5%CE%B9%CF%82_%CF%84%CE%B5%CF%81%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%BF%CF%8D

[22] Visual Studio Code – Wikipedia. Πηγή:

https://en.wikipedia.org/wiki/Visual_Studio_Code

[23] Why GitHub? 2020 GitHub, Inc. ev3dev/ vscode-ev3dev-browser, ev3dev browser for Visual Studio Code. Πηγή:

<https://github.com/ev3dev/vscode-ev3dev-browser>

[24] Python – Wikipedia. Πηγή:

<https://el.wikipedia.org/wiki/Python>

[25] Ralph Hempel et al., 2015, Python language bindings for ev3dev, python-ev3dev. Πηγή:

<https://ev3dev-lang.readthedocs.io/projects/python-ev3dev/en/stable/>

[26] EV3DEV.DRG, LEGO releases MicroPython for EV3 based on ev3dev and Pybricks. Πηγή:

<https://www.ev3dev.org/news/2019/04/13/ev3-micropython/>

[27] LEGO education, Getting started with EV3 MicroPythonVersion 1.0.0, Mar 26, 2019, CREATING AND RUNNING PROGRAMS. Πηγή:

<https://le-www-live-s.legocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf?fbclid=IwAR1JmGdaCcYr7jknKmtuoyTG8SGo4au8JkdXGvlGV6F64Pwx6XKDbMIBNzQ>

[28] Alexey Smirnov, © Alexey Smirnov 2020, Installing MicroPython on ESP32

Python vs MicroPython. Πηγή:

<https://smirnov-am.github.io/micropython-on-esp32/>

[29] Why GitHub? © 2020 GitHub, Inc., Differences Damien George edited this page on 13 Feb 2019, Differences to CPython. Πηγή:

<https://github.com/micropython/micropython/wiki/Differences>