



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΕΛΛΑΔΟΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ,
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΜΕ WORDBLOCKS ΚΑΙ
ΤΗΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON, ΜΕ
ΤΗΝ AI CAMERA HUSKYLENS ΚΑΙ ΤΟ MICRO:BIT**

**Πτυχιακή Εργασία του
Σάββα Φιλιππίδη (2657)**

Επιβλέπων: Αργύρης Κουρέας

ΣΕΡΡΕΣ, 2024

Υπεύθυνη Δήλωση: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδας.

Περίληψη

Η τεχνητή νοημοσύνη και η ρομποτική αποτελούν δύο από τους πιο συναρπαστικούς κλάδους της τεχνολογίας που επηρεάζουν και διαμορφώνουν τον σύγχρονο κόσμο. Με την ανάπτυξη καινοτόμων τεχνολογιών και την εφαρμογή τους σε πάρα πολλές πρακτικές εφαρμογές, αναδεικνύεται ολοένα και περισσότερο η σημασία τους στην καθημερινότητά μας.

Η παρούσα πτυχιακή εργασία επικεντρώνεται στη χρήση τεχνολογιών όπως το micro:bit, το Maqueen Plus και το HuskyLens, τα οποία αποτελούν εργαλεία που ενθαρρύνουν την εκμάθηση και τη δημιουργικότητα νέων και όχι μόνο στον τομέα της ρομποτικής. Μέσω πρακτικών έργων, εξερευνούμε τις δυνατότητες και τις εφαρμογές αυτών των τεχνολογιών σε πραγματικά σενάρια.

I Υλικά

- 1 Μικροϋπολογιστής micro:bit
- 1 Προγραμματιζόμενο ρομπότ Maqueen Plus V2
- 1 Κάμερα τεχνητής νοημοσύνης HuskyLens
- Παρελκόμενα (Καλώδια σύνδεσης, μεταλλικές βάσεις κτλ.)

II Μέθοδος

Σύνδεση του micro:bit με το Maqueen Plus και τοποθέτηση της κάμερας HuskyLens σε κατάλληλη θέση πάνω στο Maqueen Plus χρησιμοποιώντας τα παρελκόμενα υλικά σύμφωνα με τις οδηγίες του εγχειριδίου χρήσης που βρίσκεται στην ιστοσελίδα του κατασκευαστή <https://edu.dfrobot.com/>.

Προγραμματισμός του micro:bit στην ιστοσελίδα <https://makecode.microbit.org/> για να διαβάζει τα δεδομένα από τους αισθητήρες του Maqueen Plus, του HuskyLens και του micro:bit, ώστε να υπολογίζει βάση των δεδομένων που λαμβάνει και να στέλνει τις κατάλληλες εντολές πίσω στο εκάστοτε υλικό με βάση τα αποτελέσματα.

III Εκτέλεση έργων

Μετά τη συναρμολόγηση και τον προγραμματισμό του συστήματος σύμφωνα με την μέθοδο που περιγράφεται παραπάνω, εκτελούμε το πείραμα σε ένα ελεγχόμενο περιβάλλον (κατάλληλος φωτισμός, απουσία ηχητικών θορύβων κτλ.), παρακολουθούμε τις αντιδράσεις του συστήματος, τροποποιούμε παραμέτρους είτε στον κώδικα είτε στα υλικά του, έως ότου έχουμε το επιθυμητό αποτέλεσμα και βγάζουμε τα συμπεράσματα και τις παρατηρήσεις μας.

Περιεχόμενα

Περίληψη	4
Εισαγωγή	9
1. Τεχνητή νοημοσύνη και ρομποτική.....	10
2. Μια επισκόπηση στους μικροελεγκτές.....	11
2.1 Ο μικροελεγκτής micro:bit	12
2.2 Arduino	13
2.3 Γιατί micro:bit;.....	14
2.4 Γλώσσες προγραμματισμού στο micro:bit	15
2.5 Το Maqueen Plus	17
2.6 Η ΑΙ κάμερα HuskyLens	18
2.7 Η ρομποτική στην εκπαίδευση	20
3. Εκκίνηση με το micro:bit.....	20
3.1 Έργο 1: Εισαγωγή στο micro:bit	20
3.2 Δημιουργία νέων Έργων.....	21
3.3 Κατανόηση του περιβάλλοντος MakeCode.....	21
3.4 Δημιουργία Προγράμματος (Εναλλαγή LED μέσω ήχου)	22
3.5 Σύνδεση του micro:bit με τον Υπολογιστή.....	24
4. Σχεδιασμός έργων Maqueen Plus	25
4.1 Κατανόηση του περιβάλλοντος MakeCode για προγραμματισμό Maqueen Plus	25
4.2 Έργο 2: Κίνηση Maqueen Plus μέσω ήχου.....	27
4.3 Έργο 3: Κίνηση Maqueen Plus μέσω φωτός	31
4.4 Έργο 4: Μέτρηση απόστασης μέσω υπερήχων	34
4.5 Έργο 5: Εντοπισμός εμποδίου και παύση κίνησης.....	37
4.6 Έργο 6: Line follower με τους αισθητήρες γραμμών	40
5. Εφαρμογές Κάμερας HuskyLens.....	44
5.1 Έργο 7: Μουσική από χρώματα.....	44

5.2 Έργο 8: Line follower μέσω της κάμερας HuskyLens	58
5.3 Έργο 9: Παρακολούθηση αντικειμένου.....	63
5.4 Έργο 10: Παιχνίδι μνήμης	66
5.5 Έργο 11: Μάντεψε το χρώμα.....	72
5.6 Έργο 12: Εφέ μουσικής	75
5.7 Έργο 13: Πράσινο κόκκινο φως	77
6. Συμπεράσματα	82
Βιβλιογραφία και ιστότοποι.....	83

Πίνακας εικόνων

Εικόνα 1. Το micro:bit έκδοση v1 στα δεξιά και v2 στα αριστερά. [8].	12
Εικόνα 2. Το Arduino board (Arduino Uno). [9].....	13
Εικόνα 3. Τα blocks κώδικα WordBlocks.	16
Εικόνα 4. Το Maqueen Plus V2. [2]	18
Εικόνα 5. Η AI κάμερα HuskyLens. [2]	19
Εικόνα 6. Το προγραμματιστικό περιβάλλον MakeCode.....	21
Εικόνα 7. Κώδικας WordBlocks ‘Εναλλαγή LED μέσω ήχου’	23
Εικόνα 8. Διάγραμμα ροής ‘Εναλλαγή LED μέσω ήχου’.	24
Εικόνα 9. ‘Εναλλαγή LED μέσω ήχου’.	24
Εικόνα 10. Το προγραμματιστικό περιβάλλον MakeCode.....	26
Εικόνα 11. Το προγραμματιστικό περιβάλλον MakeCode.....	26
Εικόνα 12. Το προγραμματιστικό περιβάλλον MakeCode.....	26
Εικόνα 13. Κώδικας WordBlocks ‘Κίνηση Maqueen Plus μέσω ήχου’	28
Εικόνα 14. Διάγραμμα ροής ‘Κίνηση Maqueen Plus μέσω ήχου’.	29
Εικόνα 15. ‘Κίνηση Maqueen Plus μέσω ήχου’.	30
Εικόνα 16. ‘Κίνηση Maqueen Plus μέσω ήχου’.	30
Εικόνα 17. ‘Κίνηση Maqueen Plus μέσω ήχου’.	30
Εικόνα 18. Κώδικας WordBlocks ‘Κίνηση Maqueen Plus μέσω φωτός’	32
Εικόνα 19. Διάγραμμα ροής ‘Κίνηση Maqueen Plus μέσω φωτός’	33

Εικόνα 20. Κώδικας WordBlocks ‘Μέτρηση απόστασης μέσω υπερήχων’	34
Εικόνα 21. Διάγραμμα ροής ‘Μέτρηση απόστασης μέσω υπερήχων’	35
Εικόνα 22. Μέτρηση απόστασης μέσω υπερήχων.....	36
Εικόνα 23. Μέτρηση απόστασης μέσω υπερήχων.....	36
Εικόνα 24. Μέτρηση απόστασης μέσω υπερήχων.....	37
Εικόνα 25. Κώδικας WordBlocks ‘Έντοπισμός εμποδίου και παύση κίνησης’	38
Εικόνα 26. Διάγραμμα ροής ‘Έντοπισμός εμποδίου και παύση κίνησης’	39
Εικόνα 27. Κώδικας WordBlocks ‘Line follower με τους αισθητήρες γραμμών’	41
Εικόνα 28. Διάγραμμα ροής ‘Line follower με τους αισθητήρες γραμμών’	42
Εικόνα 29. ‘Line follower με τους αισθητήρες γραμμών’	43
Εικόνα 30. ‘Μουσική από χρώματα’	46
Εικόνα 31. ‘Μουσική από χρώματα’	47
Εικόνα 32. ‘Μουσική από χρώματα’	47
Εικόνα 33. ‘Μουσική από χρώματα’	48
Εικόνα 34. ‘Μουσική από χρώματα’	48
Εικόνα 35. ‘Μουσική από χρώματα’	49
Εικόνα 36. ‘Μουσική από χρώματα’	49
Εικόνα 37. ‘Μουσική από χρώματα’	50
Εικόνα 38. ‘Μουσική από χρώματα’	51
Εικόνα 39. ‘Μουσική από χρώματα’	51
Εικόνα 40. ‘Μουσική από χρώματα’	51
Εικόνα 41. ‘Μουσική από χρώματα’	52
Εικόνα 42. ‘Μουσική από χρώματα’	52
Εικόνα 43. ‘Μουσική από χρώματα’	52
Εικόνα 44. ‘Μουσική από χρώματα’	53
Εικόνα 45. ‘Μουσική από χρώματα’	54
Εικόνα 46. ‘Μουσική από χρώματα’	54
Εικόνα 47. ‘Μουσική από χρώματα’	54
Εικόνα 48. Κώδικας WordBlocks ‘Μουσική από χρώματα’	56
Εικόνα 49. Διάγραμμα ροής ‘Μουσική από χρώματα’	57
Εικόνα 50. Κώδικας WordBlocks ‘Line follower μέσω της κάμερας HuskyLens’	60
Εικόνα 51. Διάγραμμα ροής ‘Line follower μέσω της κάμερας HuskyLens’	62
Εικόνα 52. Κώδικας WordBlocks ‘Παρακολούθηση αντικειμένου’	64
Εικόνα 53. Διάγραμμα ροής ‘Παρακολούθηση αντικειμένου’	65

Εικόνα 54. Κώδικας WordBlocks ‘Παιχνίδι μνήμης’	69
Εικόνα 55. Διάγραμμα ροής ‘Παιχνίδι μνήμης’.	71
Εικόνα 56. Κώδικας WordBlocks ‘Μάντεψε το χρώμα’	73
Εικόνα 57. Διάγραμμα ροής ‘Μάντεψε το χρώμα’	74
Εικόνα 58. Κώδικας WordBlocks ‘Εφέ μουσικής’	76
Εικόνα 59. Διάγραμμα ροής ‘Εφέ μουσικής’.	77
Εικόνα 60. Κώδικας WordBlocks ‘Πράσινο κόκκινο φως’	80
Εικόνα 61. Διάγραμμα ροής ‘Πράσινο κόκκινο φως’.	81

Πίνακας πινάκων

Πίνακας 1. Συνοπτική σύγκριση των μικροελεγκτών micro:bit και Arduino.	15
Πίνακας 2. Συνοπτική σύγκριση μεθόδων αναγνώρισης γραμμών της κάμερα HuskyLens και των αισθητήρων γραμμών του Maqueen Plus.....	58

Εισαγωγή

Η συνεχής πρόοδος στους τομείς της τεχνητής νοημοσύνης και της ρομποτικής έχει διαμορφώσει ένα ευρύ φάσμα εφαρμογών σε πολλούς τομείς της επιστήμης και της τεχνολογίας. Από την βιομηχανία μέχρι την υγεία και την εκπαίδευση, η ενσωμάτωση τεχνολογιών όπως η τεχνητή νοημοσύνη και η ρομποτική έχει ανοίξει νέους ορίζοντες και έχει επαναπροσδιορίσει τον τρόπο που αντιλαμβανόμαστε τον κόσμο γύρω μας.

Στο πλαίσιο αυτό, η εργασία αυτή αποσκοπεί στην εξέταση της σημασίας των μικροελεγκτών στην προώθηση της επιστήμης, της τεχνολογίας και της εκπαίδευσης.

Η ενσωμάτωση μικροελεγκτών στην εκπαιδευτική ρομποτική προσφέρει στους νέους μια μοναδική ευκαιρία να εξερευνήσουν τον κόσμο της τεχνολογίας μέσα από πρακτικές εφαρμογές και πειράματα. Αυτή η προσέγγιση ενθαρρύνει την μάθηση, τη δημιουργικότητα και τη συνεργατική εργασία, ενώ παράλληλα διευρύνει το εκπαιδευτικό πεδίο πέρα από τα συνηθισμένα όρια της τάξης.

Επιπλέον, η προσβασιμότητα των μικροελεγκτών όπως το micro:bit και το Arduino καθιστούν δυνατή τη συμμετοχή ευρύτερων κοινοτήτων στην τεχνολογική και εκπαιδευτική κοινότητα. Μέσω αυτής της προσέγγισης, οι μαθητές, οι εκπαιδευτικοί και οι ερευνητές μπορούν να συμβάλλουν στη δημιουργία καινοτόμων λύσεων και να μοιραστούν τις γνώσεις τους με την παγκόσμια κοινότητα.

Η εργασία διαιρείται σε δύο βασικά τμήματα. Αρχικά, το θεωρητικό μέρος ξεκινά με μία αναφορά στην τεχνητή νοημοσύνη και την ρομποτική. Στη συνέχεια, εξετάζουμε τους μικροελεγκτές, με εστίαση στο micro:bit και την σημασία του στην εκπαίδευση, και πραγματοποιούμε μια σύγκρισή του με το δημοφιλές Arduino. Επιπρόσθετα, εξετάζουμε το Wordblocks που είναι το περιβάλλον προγραμματισμού του micro:bit καθώς και της γλώσσας Python και της JavaScript για την ανάπτυξη πιο σύνθετων έργων. Επιπλέον, εξετάζουμε το ρομποτικό κιτ Maqueen Plus καθώς και την ΑΙ κάμερα HuskyLens, εστιάζοντας σε θέματα όπως η αναγνώριση προσώπων, η παρακολούθηση αντικειμένων, η αναγνώριση γραμμών, η αναγνώριση χρωμάτων, κ.α. Τέλος, κάνουμε μια αναφορά στην σημασία της εκπαιδευτικής ρομποτικής και το S.T.E.A.M. Το δεύτερο μέρος της εργασίας επικεντρώνεται σε πειραματικές δραστηριότητες στον τομέα της εκπαιδευτικής ρομποτικής. Αυτή η πειραματική προσέγγιση περιλαμβάνει τη χρήση του μικροελεγκτή micro:bit σε συνδυασμό με την ΑΙ κάμερα HuskyLens και το ρομποτικό κιτ Maqueen Plus.

1. Τεχνητή νοημοσύνη και ρομποτική

Η τεχνητή νοημοσύνη πρόκειται για μία από τις πιο δυναμικές ερευνητικές περιοχές. Τυπικά, η αρχή της TN [1] ορίστηκε το 1956 κατά την έρευνα ορισμένων επιστημόνων, όπως ο John McCarthy, ο Marvin Minsky, ο Claude Shannon και άλλοι, αν και η έρευνα είχε ξεκινήσει νωρίτερα. Από την άλλη, η ιστορία των ρομπότ ξεκινά από την αρχαιότητα και εξελίσσεται κατά τη διάρκεια της βιομηχανικής επανάστασης. Οι πρώτες μορφές ρομπότ εμφανίστηκαν στα εργοστάσια, αυξάνοντας την παραγωγικότητα και μειώνοντας την ανθρώπινη εργασία. Στον σύγχρονο κόσμο, η ρομποτική εξελίσσεται συνεχώς παράλληλα με την τεχνητή νοημοσύνη. Τα σύγχρονα ρομπότ είναι εφοδιασμένα με τεχνητή νοημοσύνη, κάτι που τα καθιστά πιο αυτόνομα και προηγμένα.

Η τεχνητή νοημοσύνη και η ρομποτική αποτελούν δύο από τις κρίσιμες πτυχές της τεχνολογίας στον σύγχρονο κόσμο. Η επίδρασή τους επεκτείνεται σε πολλούς τομείς της κοινωνίας και της βιομηχανίας. Εκτός από τη δημιουργία ρομποτικών συστημάτων που αναλαμβάνουν εργασίες που προηγουμένως απαιτούσαν ανθρώπινη νοημοσύνη, η τεχνητή νοημοσύνη συνεπάγεται επίσης την ανάπτυξη αλγορίθμων που μπορούν να μοντελοποιήσουν και να προβλέπουν συμπεριφορές, βοηθώντας τα ρομπότ να είναι πιο αυτόνομα και αποτελεσματικά στην αλληλεπίδρασή τους με το περιβάλλον. Μια από τις κύριες προκλήσεις που αντιμετωπίζουν οι επιστήμονες είναι η δημιουργία ρομποτικών συστημάτων που να είναι ικανά να λειτουργούν εντελώς αυτόνομα, χωρίς τη συνεχή παρέμβαση ανθρώπινων χειριστών. Αυτό απαιτεί την ανάπτυξη σύνθετων αλγορίθμων που να επιτρέπουν στα ρομπότ να αντιλαμβάνονται το περιβάλλον τους, να λαμβάνουν αποφάσεις βασισμένες σε αυτήν την αντίληψη και να ενεργούν ανάλογα. Μέσω της ενσωμάτωσης αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης, τα ρομπότ αποκτούν τη δυνατότητα να προσαρμόζονται σε νέες καταστάσεις και να αντιδρούν με ακρίβεια και αποτελεσματικότητα.

Η εξέλιξη της τεχνητής νοημοσύνης και της ρομποτικής έχει οδηγήσει στην ανάπτυξη ρομποτικών συστημάτων που μπορούν να αλληλεπιδρούν με το περιβάλλον τους με πιο αποτελεσματικό τρόπο. Αυτό έχει ευρύ φάσμα εφαρμογών, από τη βιομηχανία και την υγειονομική περίθαλψη ως την εκπαίδευση και την ψυχαγωγία. Στον τομέα της υγειονομικής περίθαλψης, για παράδειγμα, οι ρομποτικές συσκευές που βασίζονται σε τεχνητή νοημοσύνη μπορούν να παρέχουν ακριβή και ασφαλή ιατρική φροντίδα, ενώ στον τομέα της εκπαίδευσης, ρομποτικά συστήματα μπορούν να λειτουργήσουν

ως εκπαιδευτικά εργαλεία, βοηθώντας τους μαθητές να κατανοήσουν δύσκολες έννοιες. Έτσι, η ενσωμάτωση της τεχνητής νοημοσύνης στη ρομποτική ανοίγει νέες δυνατότητες για τη δημιουργία προηγμένων ρομποτικών συστημάτων που μπορούν να αντιμετωπίσουν προκλήσεις σε διάφορους τομείς, προωθώντας έτσι την τεχνολογική εξέλιξη και την καινοτομία.

Η ραγδαία εξέλιξη της ρομποτικής και της τεχνητής νοημοσύνης θέτει δε ηθικές προκλήσεις που πρέπει να αντιμετωπιστούν. Μια από τις κύριες προκλήσεις είναι η ολοένα μεγαλύτερη αυτονομία των τεχνητών συστημάτων και ρομπότ. Καθώς αυτά γίνονται όλο και πιο ικανά να λαμβάνουν αποφάσεις χωρίς ανθρώπινη παρέμβαση, προκύπτουν ερωτήματα σχετικά με την ευθύνη και την ηθική τους συμπεριφορά. Επιπλέον, υπάρχει η ανησυχία για την αντικατάσταση ανθρώπινης εργασίας από ρομπότ, η οποία μπορεί να οδηγήσει σε ανεργία και ανισότητα. Άλλα ηθικά ζητήματα περιλαμβάνουν την ιδιωτικότητα και την ασφάλεια δεδομένων, καθώς και την πιθανή δημιουργία ηθικών διλημμάτων σε καταστάσεις όπου η τεχνητή νοημοσύνη πρέπει να πάρει αποφάσεις με κρίσιμες επιπτώσεις, όπως στον τομέα της υγείας ή της δικαιοσύνης. Επομένως, η διασφάλιση της δεοντολογίας και της διαφάνειας στην ανάπτυξη και χρήση τέτοιων τεχνολογιών είναι ουσιώδης για τη δημιουργία ενός κοινωνικά αποδεκτού και υπεύθυνου περιβάλλοντος.

2. Μια επισκόπηση στους μικροελεγκτές

Οι μικροελεγκτές αποτελούν βασικά εργαλεία στον κόσμο της ρομποτικής, της ηλεκτρονικής και του Internet of Things (IoT). Πρόκειται για μικρούς υπολογιστές που σχεδιάστηκαν για να εκτελούν ειδικές λειτουργίες σε ηλεκτρονικά συστήματα. Συνήθως χρησιμοποιούνται σε εφαρμογές όπου απαιτείται έλεγχος και διαχείριση υλικού ή εξειδικευμένων διαδικασιών σε περιορισμένο περιβάλλον, όπως στα συστήματα έξυπνου σπιτιού, στα αυτοκίνητα, σε βιομηχανικές εφαρμογές, σε φορητές συσκευές κτλ. Οι μικροελεγκτές συνήθως διαθέτουν περιορισμένους πόρους σε σχέση με έναν κανονικό υπολογιστή, όπως περιορισμένη μνήμη και επεξεργαστική ισχύ. Ωστόσο, αυτό τους καθιστά ιδανικούς για εφαρμογές που απαιτούν χαμηλή κατανάλωση ενέργειας, μικρό μέγεθος και χαμηλό κόστος.

2.1 Ο μικροελεγκτής micro:bit

Ο micro:bit, πρόκειται για έναν μικροελεγκτή ο οποίος σχεδιάστηκε από το BBC και έχει επιδείξει την ικανότητά του να εισάγει τους νέους και όχι μόνο στον κόσμο της τεχνολογίας και του προγραμματισμού. Με αισθητήρες, εύκολη προσβασιμότητα και εκπαιδευτικό υλικό, αποτελεί ένα πολύτιμο εργαλείο για την εκμάθηση και τη δημιουργία.



Εικόνα 1. Το micro:bit έκδοση v1 στα δεξιά και v2 στα αριστερά. [8].

Οι διαστάσεις του είναι μικρότερες από μία τραπεζική κάρτα, ενώ διαθέτει επεξεργαστή ARM Cortex-M0, αισθητήρες επιτάχυνσης και μαγνητόμετρου, συνδεσιμότητα Bluetooth και USB, μία οθόνη η οποία αποτελείται από 25 LEDs και δύο προγραμματιζόμενα κουμπιά. Μπορεί επίσης να συνδεθεί με άλλες συσκευές όπως ηλεκτρονικά κυκλώματα, αισθητήρες και ηλεκτρονικά περιφερειακά. Αυτά τα χαρακτηριστικά το καθιστούν ιδανικό για διάφορα προγραμματιστικά έργα και εφαρμογές.

Οι χρήστες μπορούν να προγραμματίσουν το micro:bit χρησιμοποιώντας είτε το εύκολο στη χρήση περιβάλλον Wordblocks, είτε τις γλώσσες προγραμματισμού Python και JavaScript. Αυτή η ευελιξία επιτρέπει σε χρήστες όλων των επιπέδων να εξερευνήσουν τον κόσμο του προγραμματισμού. Ο micro:bit δεν παρέχει μόνο τα εργαλεία για τη δημιουργία, αλλά και πλούσιο εκπαιδευτικό υλικό και παιχνίδια μέσω του διαδικτύου. Ο σκοπός του είναι να ενθαρρύνει την ενεργό συμμετοχή των παιδιών στον κόσμο της τεχνολογίας και του προγραμματισμού, προωθώντας τη δημιουργικότητά τους.

Το micro:bit αντιπροσωπεύει κάτι περισσότερο από ένα απλό εργαλείο προγραμματισμού. Η ευκολία χρήσης, τα πλούσια εκπαιδευτικά περιεχόμενα και η δυνατότητα προσαρμογής με άλλα περιφερειακά και αισθητήρες, ανοίγει νέους

ορίζοντας στην εκπαίδευση και τη δημιουργικότητα, ενθαρρύνοντας τους χρήστες να εξερευνούν και να δημιουργούν.

2.2 Arduino

Το Arduino πρόκειται για έναν μικροελεγκτή που κατασκευάστηκε από την ομώνυμη εταιρία και είναι ίσως ο δημοφιλέστερος στο είδος του. Τα Arduino boards μπορούν να διαβάζουν εισόδους από αισθητήρες και να τις μετατρέπουν σε εξόδους, όπως ενεργοποίηση κινητήρων ή LED και άλλων υλικών και περιφερειακών. Οι χρήστες προγραμματίζουν το Arduino χρησιμοποιώντας τη γλώσσα προγραμματισμού Arduino που βασίζεται στις γλώσσες C/C++ και το Λογισμικό Arduino IDE, βασισμένο στην πλατφόρμα Processing. Η εξέλιξη του Arduino οφείλεται σε μια παγκόσμια κοινότητα δημιουργών που προσθέτουν γνώσεις και δυνατότητες στην πλατφόρμα. Γεννήθηκε στο Ινστιτούτο Αλληλεπίδρασης Inrea για ταχεία προτυποποίηση και εκπαίδευση στην ρομποτική και προγραμματισμό, αλλά εξελίχθηκε σε προϊόντα που καλύπτουν νέες ανάγκες όπως εφαρμογές IoT και φορητές συσκευές.

Μέσα στα χρόνια, το Arduino έχει αποτελέσει τον εγκέφαλο χιλιάδων έργων, από καθημερινά αντικείμενα έως και πολύπλοκα επιστημονικά όργανα. Μια παγκόσμια κοινότητα δημιουργών - φοιτητές, ερασιτέχνες, καλλιτέχνες, προγραμματιστές και επαγγελματίες - έχει συγκεντρωθεί γύρω από αυτήν την ανοικτή πλατφόρμα, με τις συνεισφορές τους να προσθέτουν μια απίστευτη ποσότητα προσιτών γνώσεων που μπορεί να είναι μεγάλη βοήθεια τόσο για αρχάριους όσο και για ειδικούς.



Εικόνα 2. Το Arduino board (Arduino Uno). [9].

2.3 Γιατί micro:bit;

Όπως προαναφέρθηκε ο μικροελεγκτής micro:bit διαθέτει οθόνη 5X5 LED φώτων καθώς και μία γκάμα από ενσωματωμένους αισθητήρες όπως, πλήκτρα πίεσεως, μαγνητόμετρο, επιταχυνσιόμετρο, αισθητήρα φωτός καθώς επίσης λειτουργία Bluetooth, καθώς και ακροδέκτες εισόδου/εξόδου και μικρότερους ακροδέκτες για περισσότερες δυνατότητες. Γενικά, το micro:bit έχει περισσότερη ισχύ επεξεργασίας και μνήμη από το Arduino. Ο κύριος λόγος που μπορεί να επιλέξει κάποιος το micro:bit είναι ότι η οθόνη LED του μπορεί να λειτουργήσει ως οπτική οθόνη καθαυτή και μπορείτε να αλληλεπιδράτε με αυτήν μέσω των πλήκτρων στα πλάγια του.

Ως αποτέλεσμα, δεν χρειάζεται να δαπανηθούν περεταίρω χρήματα ή να κατασκευαστούν πολύπλοκες συσκευές για να γίνει προγραμματισμός η απασφαλμάτωση ενός micro:bit. Είναι αρκετά φιλικό προς τους αρχάριους, και το BBC το σχεδίασε με γνώμονα την εκπαίδευση νεαρών μαθητών στο σχολείο. Οι γλώσσες προγραμματισμού που είναι συμβατές με το micro:bit περιλαμβάνουν την Python και τη JavaScript, που θεωρούνται ιδιαίτερα εύκολες στην εκμάθηση για τον αρχάριο χρήστη, καθώς επίσης και το περιβάλλον Wordblocks το οποίο βασίζεται στον οπτικό προγραμματισμό και επικεντρώνεται στο να ξεκινήσουν οι νέοι μικρής ηλικίας τον προγραμματισμό. Επίσης, σημαντικό είναι να σημειωθεί ότι το περιβάλλον ανάπτυξης του micro:bit είναι βασισμένο στον ιστό, οπότε δεν χρειάζεται να κατεβάσουμε και να εγκαταστήσουμε τίποτα για να το προγραμματίσουμε.

Από την άλλη, ο βασικός σχεδιασμός του Arduino επικεντρώνεται στους πολλούς ακροδέκτες εισόδου/εξόδου που διαθέτει κατά μήκος των ακμών του. Αυτοί παρουσιάζουν απεριόριστες δυνατότητες για σύνδεση με άλλα ηλεκτρονικά εξαρτήματα, αλλά αυτό απαιτεί τόσο την δαπάνη περισσότερων χρημάτων σε υλικό, καθώς και πιο σύνθετες συνδεσμολογίες ή προβλήματα συμβατότητας μεταξύ των εκατοντάδων συσκευών του εμπορίου, εκδόσεων υλικού κτλ. Για να προγραμματίσουμε το Arduino, πρέπει να κατεβάσουμε το περιβάλλον ανάπτυξης Arduino (IDE), καθώς και να αναζητήσουμε βιβλιοθήκες υλικού για τις εκάστοτε περιφερειακές συσκευές, διαδικασίες οι οποίες είναι χρονοβόρες και μπορεί να αποθαρρύνουν τον αρχάριο χρήστη ή να δημιουργήσουν προβλήματα σε μια τάξη όπου διδάσκεται η εκπαιδευτική ρομποτική.

Μέσα στο περιβάλλον ανάπτυξης, οι γλώσσες που μπορούμε να χρησιμοποιήσουμε είναι η C και η C++, οι οποίες ομολογουμένως χρειάζονται μια εξοικείωση ως έναν

βαθμό για να τις χρησιμοποιούμε αποτελεσματικά. Η έμφαση στους ακροδέκτες GPIO και η κάπως πιο πολύπλοκη διαδικασία προγραμματισμού καθιστούν το Arduino καλύτερο για εκείνους που επιθυμούν να εξερευνήσουν βαθύτερα τη δημιουργία ηλεκτρονικών και ρομποτικών έργων.

Κριτήριο	micro:bit	Arduino
Οθόνη LED	Οθόνη LED 5x5	-
Πλήκτρα πίεσεως	Ναι	-
Επιταχυνσιόμετρο	Ναι	-
Λειτουργία Bluetooth	Ναι	-
Γλώσσες προγραμματισμού	WordBlocks, Python και JavaScript	C/C++
Ευκολία χρήσης	Σχεδιάστηκε για εκπαιδευτική χρήση	Πιο προχωρημένο
Πληροφορικό υλικό	Διαθέτει πλούσιο εκπαιδευτικό υλικό	-

Πίνακας 1. Συνοπτική σύγκριση των μικροελεγκτών micro:bit και Arduino.

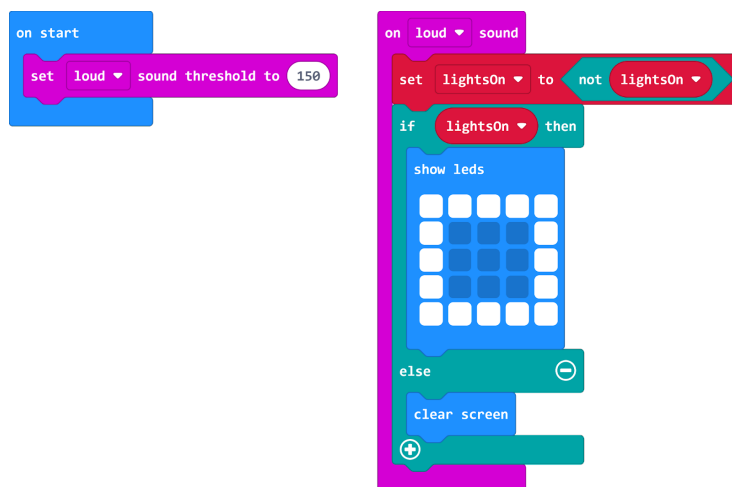
Συμπεραίνουμε λοιπόν, πως το micro:bit προκύπτει ως μια εξαιρετική επιλογή για αρχάριους καθώς και για εκπαιδευτικούς σκοπούς. Η εύκολη χρήση, η οθόνη LED και η διαθεσιμότητα πλούσιου εκπαιδευτικού υλικού το καθιστούν ιδανικό για την εισαγωγή στον κόσμο της ρομποτικής και του προγραμματισμού, ενώ παράλληλα παρέχει επαρκή ευελιξία για πειραματισμούς και δημιουργικότητα.

2.4 Γλώσσες προγραμματισμού στο micro:bit

Ως ένα εκπαιδευτικό εργαλείο το οποίο επικεντρώνεται στους νέους, το micro:bit χρησιμοποιεί το προγραμματιστικό περιβάλλον WordBlocks. Το WordBlocks σχετίζεται με τον οπτικό προγραμματισμό στον οποίο οι εντολές κώδικα αναπαρίστανται ως εικονίδια που μπορούν να σύρονται και να τοποθετούνται σε ένα πλέγμα σαν τουβλάκια. Ο προγραμματισμός μέσω αυτών των εικονιδίων/blocks κώδικα, επιτρέπει στα παιδιά που είναι στα πρώτα τους βήματα με τον προγραμματισμό και την ρομποτική να κατανοήσουν και να εμπεδώσουν βασικές αρχές προγραμματισμού, με διαδραστικό και ευχάριστο τρόπο, χωρίς την ανάγκη των να μάθουν πρώτα τη σύνταξη μίας ολοκληρωμένης γλώσσας. Τα παιδιά μπορούν να χειριστούν ένα ποντίκι σε πιο νεαρή ηλικία, έτσι ο προγραμματισμός με WordBlocks αφαιρεί σχεδόν ολοκληρωτικά την πληκτρολόγηση από την εξίσωση.

Ενώ ο προγραμματισμός με WordBlocks έχει πολλά πλεονεκτήματα για τα παιδιά και τους αρχάριους, υπάρχουν κάποια μειονεκτήματα σε σύγκριση με μια ολοκληρωμένη

γλώσσα προγραμματισμού βασισμένη στον γραπτό κώδικα. Οποιοσδήποτε θέλει να εμβυθύνει στον προγραμματισμό, δεν μπορεί να στηρίζεται μόνο σε γλωσσικά περιβάλλοντα προγραμματισμού όπως το WordBlocks. Ειδικά εάν κάποιος θέλει να προγραμματίσει πιο σύνθετες εφαρμογές, η μετάβαση σε κάποια ολοκληρωμένη γλώσσα προγραμματισμού είναι απαραίτητη.



Εικόνα 3. Τα blocks κώδικα WordBlocks.

Εκτός από το WordBlocks, οι χρήστες μπορούν να προγραμματίσουν το micro:bit και χρησιμοποιώντας τη γλώσσα προγραμματισμού Python καθώς και την JavaScript. Αυτές οι γλώσσες προσφέρουν έναν πιο προηγμένο προγραμματισμό σε σύγκριση με το WordBlocks, επιτρέποντας στους χρήστες να δημιουργήσουν πιο πολύπλοκες εφαρμογές με μεγαλύτερη ευελιξία. Η Python [10] αναδεικνύεται ως μια προγραμματιστική γλώσσα υψηλού επιπέδου, η οποία εστιάζει στην ευανάγνωστη κωδικοποίηση μέσω της χρήσης εσοχών για την μορφοποίηση του κώδικα. Υποστηρίζει ποικίλες μεθόδους προγραμματισμού, συμπεριλαμβανομένων της δομημένης και της αντικειμενοστραφούς προσέγγισης. Η Python κατατάσσεται ανάμεσα στις πιο δημοφιλείς γλώσσες προγραμματισμού και έχει εκτεταμένη χρήση στον χώρο της μηχανικής μάθησης.

Η JavaScript [11] είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία συναντάται κυρίως στη δημιουργία δυναμικών και αλληλεπιδραστικών ιστοσελίδων. Έχει δυνατότητες όπως η αντικειμενοστραφής προσέγγιση, και παρέχει δυναμικούς

τύπους δεδομένων. Τα περισσότερα προγράμματα περιήγησης διαθέτουν μηχανισμούς για την εκτέλεση JavaScript, ενώ έχει επεκταθεί και σε διακομιστές μέσω βιβλιοθηκών όπως το Node.js. Λόγω της σχετικά εύκολης εκμάθησης και χρήσης της, χρησιμοποιείτε και σε εφαρμογές πέρα από το διαδίκτυο, όπως στην εκπαιδευτική ρομποτική με το micro:bit.

Καταλήγοντας, το WordBlocks προσφέρει ένα εκπαιδευτικό εργαλείο για νέους, επιτρέποντας τους να εξοικειωθούν με τον προγραμματισμό μέσω οπτικών εικονιδίων. Ενώ αυτή η μέθοδος είναι εύκολη και ενδιαφέρουσα για αρχάριους, η μετάβαση σε μια ολοκληρωμένη γλώσσα προγραμματισμού είναι αναπόφευκτη για πιο σύνθετες εφαρμογές. Τόσο η Python όσο και η JavaScript προσφέρουν αυτήν την ευελιξία, επιτρέποντας στους χρήστες να αναπτύξουν πιο προηγμένες εφαρμογές και να εμβαθύνουν στον προγραμματισμό. Ως εκ τούτου, το micro:bit με το WordBlocks αποτελεί μια εξαιρετική εισαγωγή στον κόσμο του προγραμματισμού, ενώ η δυνατότητα χρήσης της Python και της JavaScript παρέχει μια πιο προηγμένη προοπτική για τους πιο έμπειρους χρήστες.

2.5 Το Maqueen Plus

Το Maqueen Plus πρόκειται για ένα εκπαιδευτικό ρομπότ που σχεδιάστηκε για να διδάσκει βασικές αρχές της ρομποτικής σε νέους. Με μία πληθώρα αισθητήρων, κινητήρων και λειτουργιών, το Maqueen Plus παρέχει μια διαδραστική εμπειρία εκπαίδευσης, προάγοντας την ανάπτυξη δεξιοτήτων στον τομέα της ρομποτικής και του STEM. Σε συνδυασμό με το micro:bit, το Maqueen Plus επιτρέπει στα παιδιά να δημιουργήσουν διάφορα έργα και να εξερευνήσουν τις δυνατότητες της ρομποτικής, κάνοντας τη μάθηση διασκεδαστική και ενδιαφέρουσα.

Το Maqueen Plus αποτελεί ιδανική επιλογή για εκπαιδευτικά προγράμματα σχολείων και κέντρων εξωσχολικών δραστηριοτήτων. Το ρομπότ διαθέτει αισθητήρα υπερήχων, μοτέρ για κίνηση, LEDs, αισθητήρες ανίχνευσης γραμμών, υποδοχή για micro:bit κ.α. Υποστηρίζει πολλές επιλογές ενέργειας και παρέχει θύρες επέκτασης για επιπλέον περιφερειακά. Είναι εύκολο στη συναρμολόγηση και τον προγραμματισμό, καθιστώντας το κατάλληλο ακόμα και για αρχάριους.



Εικόνα 4. Το Maqueen Plus V2. [2]

2.6 Η ΑΙ κάμερα HuskyLens

Η κάμερα HuskyLens αποτελεί ένα εκπαιδευτικό εργαλείο που ενσωματώνει πολλές λειτουργίες τεχνητής νοημοσύνης, επιτρέποντας σε ρομποτικά συστήματα να αντιλαμβάνονται το περιβάλλον και βάση αυτού να λαμβάνουν αποφάσεις.

Περιλαμβάνει έναν σημαντικό αριθμό από λειτουργίες που καλύπτουν ένα ευρύ φάσμα εφαρμογών:

- Αναγνώριση Προσώπων (Face Recognition): Η δυνατότητα της κάμερας HuskyLens να αναγνωρίζει πρόσωπα συναντάται σε πολλές εφαρμογές, όπως αυτοματοποιημένα συστήματα ασφαλείας αναγνώρισης προσώπων, για πρόσβαση σε εγκαταστάσεις ή σε προσωπικές ηλεκτρονικές συσκευές όπως τα κινητά τηλέφωνα.
- Παρακολούθηση Αντικειμένων (Object Tracking): Η κάμερα HuskyLens μπορεί να ανιχνεύει και να παρακολουθεί διάφορα αντικείμενα, πρόσωπα, ζώα κτλ. Αυτή η τεχνολογία είναι πολύ χρήσιμη στην καθημερινότητα, με εφαρμογές όπως ασφάλεια χώρων, έξυπνα drones κ.τ.λ.
- Αναγνώριση Αντικειμένων (Object Recognition): Με την αναγνώριση αντικειμένων δίνεται η δυνατότητα να αναγνωρίζονται διάφορα αντικείμενα, όπως μπουκάλια, βιβλία, ή οποιαδήποτε άλλο αντικείμενο. Εφαρμογές της τεχνολογίας αυτής μπορεί να εφαρμοστούν σε βιομηχανικές μηχανές αυτοματοποιημένου ελέγχου αποθήκευσης ή και ποιοτικού ελέγχου.

- Ανίχνευση Γραμμών (Line Tracking): Η ανίχνευση είναι σημαντική για πολλές ρομποτικές εφαρμογές, όπως οι ακολουθητές γραμμών (line followers) σε αυτοκινούμενα ρομπότ ή σε βιομηχανικά συστήματα μεταφοράς.
- Αναγνώριση Χρωμάτων (Color Recognition): Πολύ χρήσιμη επίσης είναι η αναγνώριση χρωμάτων η οποία μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές, όπως η αναγνώριση εμποδίων ή καταμέτρηση αντικειμένων σε ένα περιβάλλον παραγωγής.
- Αναγνώριση Ετικετών (Tag Recognition): Η αναγνώριση ετικετών έχει χρήση σε πάρα πολλές εφαρμογές, όπως η διαχείριση αποθηκών ή η αναγνώριση συσκευασιών σε βιομηχανικό επίπεδο.
- Αναγνώριση QR και Barcodes: Η δυνατότητα αναγνώρισης QR και barcodes την οποία συναντάμε πολύ συχνά στην καθημερινότητα μας, όπως για παράδειγμα κατά την επίσκεψη μας σε κάποιο πολυκατάστημα και αφορά την αυτόματη αναγνώριση προϊόντων και την κοστολόγηση τους ή στην παρακολούθηση αντικειμένων σε μια αλυσίδα εφοδιασμού.

Αυτές οι λειτουργίες επιτρέπουν στην κάμερα HuskyLens να εφαρμόζει πολλές τεχνικές τεχνητής νοημοσύνης για να αντιληφθεί και να αλληλεπιδράσει με το περιβάλλον του, ενσωματώνοντας έτσι την εξέλιξη της τεχνολογίας στη ρομποτική και την τεχνητή νοημοσύνη. Σε συνδυασμό με το micro:bit το οποίο λειτουργεί ως το τμήμα επεξεργασίας των δεδομένων που λαμβάνει ή κάμερα, καθώς και με τις δυνατότητες κίνησης του Maqueen Plus, έχουμε την δυνατότητα να υλοποιήσουμε μια ευρεία γκάμα έργων.



Εικόνα 5. Η AI κάμερα HuskyLens. [2]

2.7 Η ρομποτική στην εκπαίδευση

Η ενσωμάτωση της ρομποτικής στην εκπαίδευση προσφέρει νέες ευκαιρίες για την ανάπτυξη δεξιοτήτων στους μαθητές. Βασίζεται σε αρχές [13] όπως αυτές του Piaget και του Papert, που υπογραμμίζουν τη σημασία της προσαρμογής της συμπεριφοράς μέσω της εμπειρίας και της αλληλεπίδρασης με το περιβάλλον. Μέσω της δοκιμής και του λάθους, καθώς και της συνεργατικότητας, οι μαθητές αναπτύσσουν κοινωνικές δεξιότητες και μαθαίνουν μέσα από τα λάθη τους, ενισχύοντας την ενεργή μάθηση και την αυτοδιορθωτική τους διαδικασία. Συνεπώς, η ενσωμάτωση της ρομποτικής στην εκπαίδευση προωθεί μια προσέγγιση που ενισχύει τη δημιουργικότητα, την προσαρμογή και την επίλυση προβλημάτων στο πλαίσιο της εκπαιδευτικής διαδικασίας.

Μέσω των δραστηριοτήτων S.T.E.A.M[12] (Επιστήμη, Τεχνολογία, Μηχανική, Τέχνη, Μαθηματικά), η εκπαίδευση προετοιμάζει τους νέους για τη σύγχρονη εποχή.

Με εφαρμογές και κατασκευές, οι μαθητές αποκτούν πρακτικές εμπειρίες που τους εξοικειώνουν με την τεχνολογία και την επιστήμη, ενθαρρύνοντας το ενδιαφέρον και την αυτοπεποίθησή τους.

Συνολικά, η ρομποτική στην εκπαίδευση ανοίγει νέους ορίζοντες για την ανάπτυξη των μαθητών και την εξοικείωσή τους με τις νέες τεχνολογίες. Αποτελεί ένα εκπαιδευτικό εργαλείο που ενδυναμώνει τους μαθητές και τους προετοιμάζει για μία δημιουργική και επιστημονική πορεία στο μέλλον.

3. Εκκίνηση με το micro:bit

Το micro:bit πρόκειται για έναν μικροϋπολογιστή που σχεδιάστηκε για να εισαγάγει νέους και μη χρήστες στον κόσμο της ρομποτικής και του προγραμματισμού. Με ενσωματωμένα εξαρτήματα όπως LED οθόνη, αισθητήρες κίνησης και επιταχυνσιόμετρο, παρέχει ένα ιδανικό περιβάλλον για την εκμάθηση βασικών αρχών προγραμματισμού.

3.1 Έργο 1: Εισαγωγή στο micro:bit

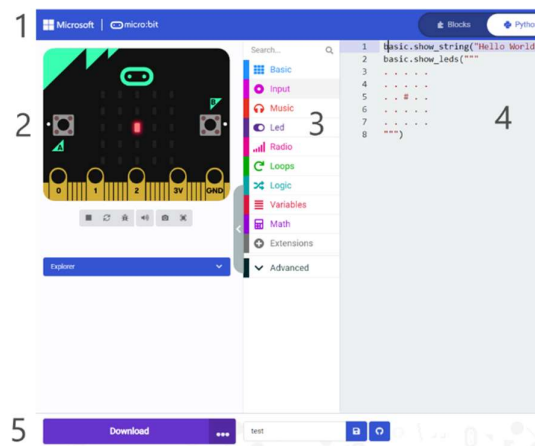
Θα χρησιμοποιήσουμε ένα εργαλείο προγραμματισμού που ονομάζεται MakeCode. Ανοίγουμε τον web browser και πλοηγούμαστε στην ιστοσελίδα <https://makecode.micro:bit.org/>.

Αυτή η πλατφόρμα θα είναι το περιβάλλον προγραμματισμού μας, όπου θα προγραμματίσουμε και θα δημιουργήσουμε προγράμματα για το micro:bit.

3.2 Δημιουργία νέων Έργων

1. Κάνουμε κλικ σε "New Project" για να ξεκινήσουμε το πρώτο μας έργο.
2. Δίνουμε ένα όνομα στο έργο, για παράδειγμα "Project 1".

3.3 Κατανόηση του περιβάλλοντος MakeCode



Εικόνα 6. Το προγραμματιστικό περιβάλλον MakeCode

1. Μενού

Διαθέτει διάφορες λειτουργίες όπως πρόσβαση στις ρυθμίσεις του περιβάλλοντος και άλλες με πιο σημαντική την εναλλαγή μεταξύ γλωσσών προγραμματισμού Blocks, JavaScript και Python.

2. Προσομοιωτής micro:bit

Ο προσομοιωτής λειτουργεί αυτόματα μετά από μια αλλαγή στον κώδικα. Τα κουμπιά κάτω από τον προσομοιωτή επιτρέπουν να σταματήσουμε/ξεκινήσουμε την εκτέλεση του προγράμματος, να ενεργοποιήσουμε την λειτουργία αποσφαλμάτωσης κ.α.

3. Εργαλειοθήκη

Συναρτήσεις και άλλες χρήσιμες λειτουργίες οι οποίες συνθέτουν τα προγράμματα του micro:bit.

4. Επεξεργαστής κειμένου

Εδώ γράφουμε τον κώδικα για τα προγράμματα micro:bit.

5. Εργαλειοθήκη

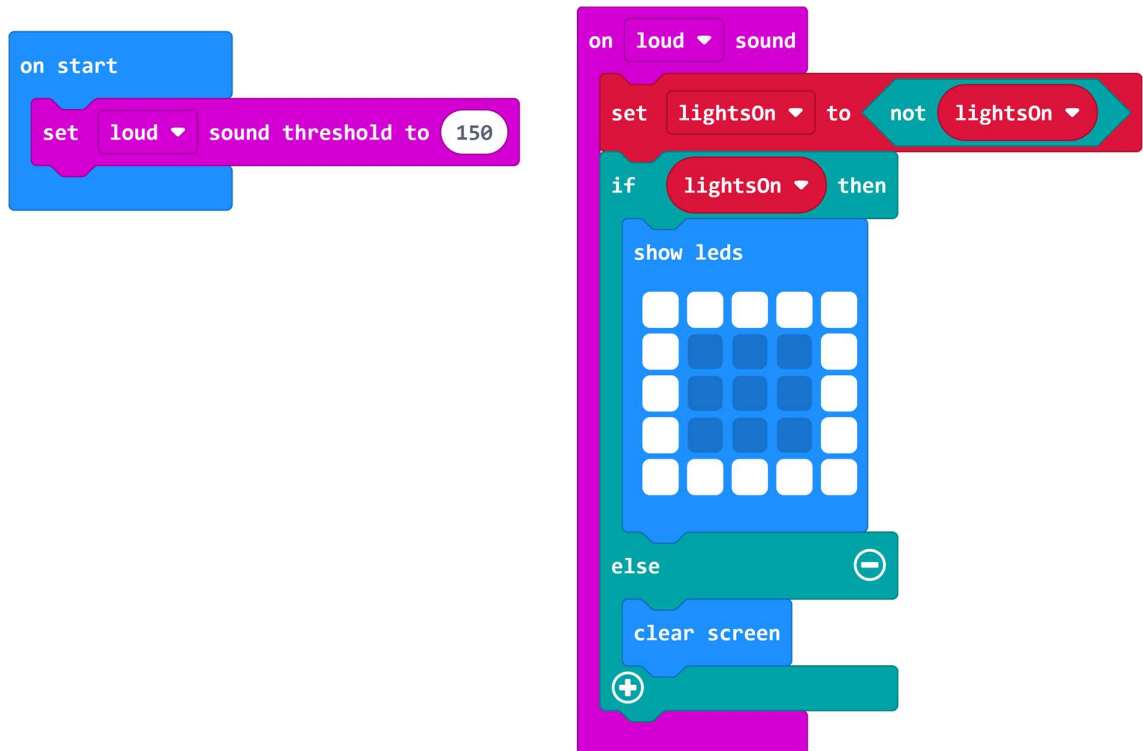
Διάφορες λειτουργίες με σημαντικότερο το πλήκτρο Λήψης με το οποίο το οποίο μεταγλωττίζουμε και μεταφέρουμε το πρόγραμμα μας στη συσκευή micro:bit.

3.4 Δημιουργία Προγράμματος (Εναλλαγή LED μέσω ήχου)

Κώδικας και τεκμηρίωση

```
lightsOn = False

def on_sound_loud():
    global lightsOn
    lightsOn = not (lightsOn)
    if lightsOn:
        basic.show_leds("""
            # # # # #
            # . . . #
            # . . . #
            # . . . #
            # # # # #
            """)
    else:
        basic.clear_screen()
input.on_sound(DetectedSound.LOUD, on_sound_loud)
input.set_sound_threshold(SoundThreshold.LOUD, 150)
```

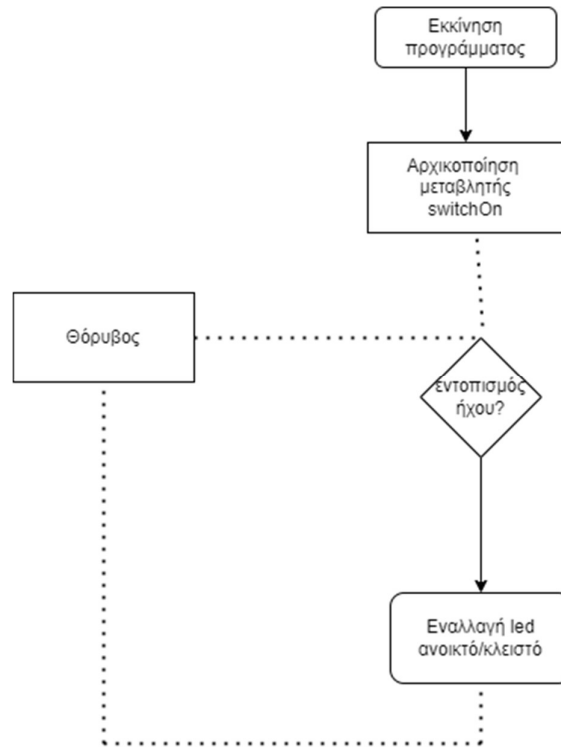


Εικόνα 7. Κώδικας WordBlocks 'Εναλλαγή LED μέσω ήχου'

Τεκμηρίωση κώδικα:

Με την εκτέλεση του κώδικα οι λαμπτήρες LED του micro:bit σβήνουν ή ανάβουν μέσω ενός δυνατού ήχου στο μικρόφωνο την ευαισθησία του οποίου μπορούμε να ρυθμίσουμε από το κατώφλι ευαισθησίας.

- Η lights_on είναι μια μεταβλητή που χρησιμοποιείται για να παρακολουθεί την κατάσταση των LED.
- Η συνάρτηση on_sound_loud() εναλλάσσει την κατάσταση της lights_on και ανάβει ή σβήνει τα LED ανάλογα την τιμή της.
- Ο ακροατής συμβάντος (input.on_sound) ρυθμίζεται να καλεί την on_sound_loud όταν εντοπιστεί ένας δυνατός ήχος.
- Το κατώφλι ήχου για την ανίχνευση δυνατού ήχου ορίζεται στο 150 χρησιμοποιώντας το input.set_sound_threshold.

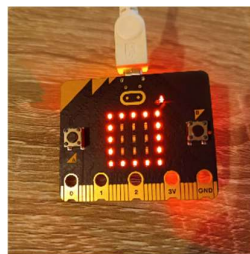


Εικόνα 8. Διάγραμμα ροής 'Εναλλαγή LED μέσω ήχου'.

3.5 Σύνδεση του micro:bit με τον Υπολογιστή

1. Συνδέσουμε το micro:bit με τον υπολογιστή χρησιμοποιώντας το καλώδιο USB.
2. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικά στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.
3. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.
4. Δοκιμή στο micro:bit:

Πατάμε το πλήκτρο επαναφοράς στο micro:bit για να τρέξουμε το πρόγραμμά. Κάνοντας έναν δυνατό ήχο (π.χ. παλαμάκι) διαπιστώνουμε ότι η οθόνη του micro:bit θα σβήσει (εικ. 1). Επαληθεύοντας τον ήχο, η οθόνη θα ανάψει (εικ. 2). Διαπιστώνουμε ότι ο ήχος λειτουργεί ως ένας διακόπτης φωτός εξ αποστάσεως.



Εικόνα 9. 'Εναλλαγή LED μέσω ήχου'.

Παρατηρήσεις

Η ευαισθησία του μικροφώνου του micro:bit εξαρτάται από την τιμή που θα ορίσουμε ως κατώφλι, οπότε με μικρότερη τιμή θα είναι ευκολότερο να ανάψουν/σβήσουν τα LEDs, όμως θα είναι επίσης ευκολότερο να συμβεί αυτό από λάθος, όπως π χ θόρυβος στο περιβάλλον.

4. Σχεδιασμός έργων Maqueen Plus

Αυτό το κεφάλαιο παρουσιάζει το Maqueen Plus και εξετάζει τις δυνατότητες που προσφέρει στην εκπαίδευση. Καλύπτονται οι διαδικασίες προετοιμασίας του ρομπότ και παρέχονται οδηγίες για την εκτέλεση διαφορετικών διεργασιών και προγραμματισμό.

Το Maqueen Plus είναι ένα προγραμματιζόμενο ρομπότ που σχεδιάστηκε για την εκπαίδευση στην τεχνολογία και την ρομποτική. Μέσω των παρελκόμενων αισθητήρων που διαθέτει καθώς και με την δυνατότητα επέκτασης με μια πληθώρα εξαρτημάτων το Maqueen Plus είναι ένα πολύπλευρο εργαλείο για την εκπαίδευση στον τομέα της ρομποτικής.

4.1 Κατανόηση του περιβάλλοντος MakeCode για προγραμματισμό Maqueen Plus

1. Ανοίγουμε τον web browser και προηγούμεστε στη σελίδα <https://makecode.micro:bit.org/>.

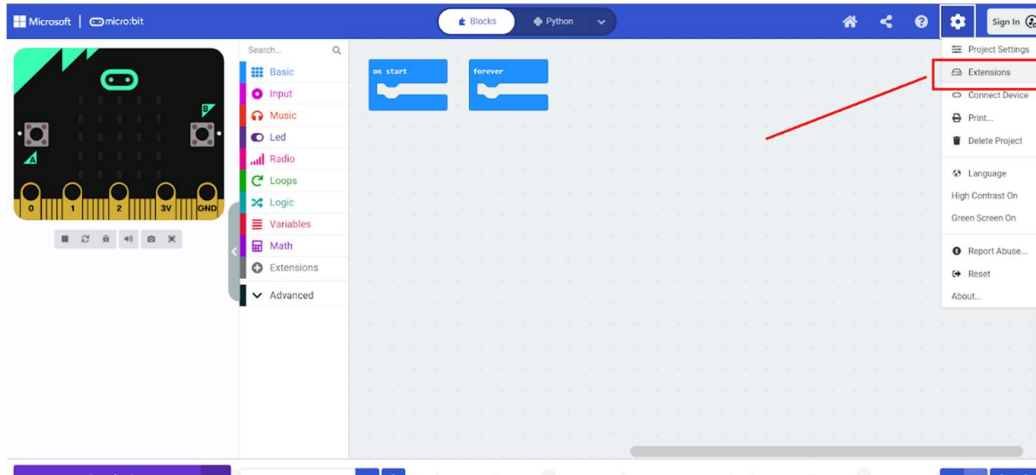
2. Κάνουμε κλικ σε "New Project" για να φτιάξουμε ένα καινούριο έργο και ορίζουμε ένα όνομα της επιλογής μας και το ανοίγουμε.

3. Σε αυτό το σημείο θα χρειαστεί να εισάγουμε την βιβλιοθήκη που είναι υπεύθυνη για το Maqueen Plus. Ανοίγουμε τις ρυθμίσεις(γρανάζι) και πατάμε στην επιλογή "Extensions" *[Εικόνα 10]*

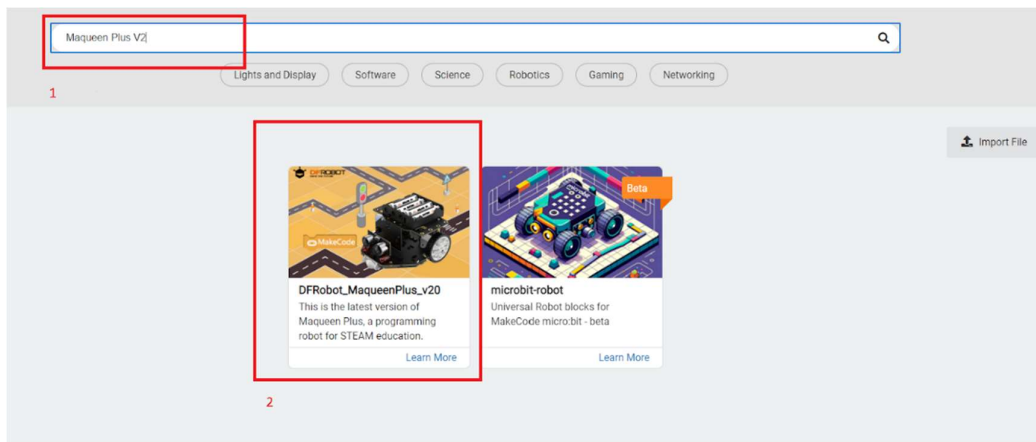
4. Στην συνέχεια γράφουμε στο πεδίο αναζήτησης "Maqueen Plus V2" και πατώντας enter θα μας εμφανίσει το "DFRobot_MaqueenPlus_v20" το οποίο και επιλέγουμε και αυτομάτως θα ακολουθήσει η εγκατάσταση της βιβλιοθήκης. *[Εικόνα 11]*

5. Σε αυτό το σημείο η εγκατάσταση της βιβλιοθήκης θα πρέπει να έχει ολοκληρωθεί επιτυχώς και θα βλέπουμε την επιλογή maqueenPlusV2 στην εργαλειοθήκη του MakeCode όπως στην παρακάτω εικόνα. *[Εικόνα 12]*

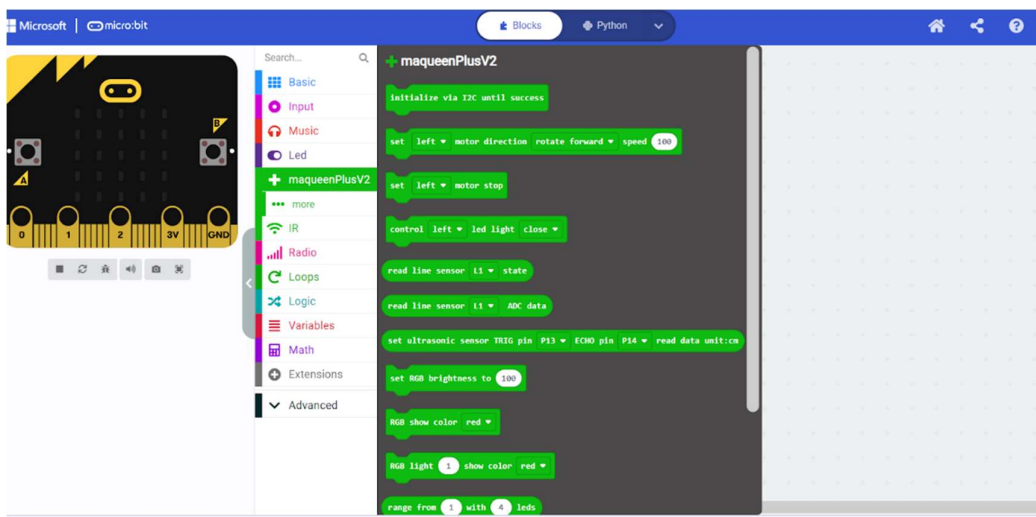
ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΜΕ WORDBLOCKS ΚΑΙ ΤΗΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON, ΜΕ ΤΗΝ AI CAMERA HUSKYLENS ΚΑΙ ΤΟ MICRO:BIT



Εικόνα 10. Το προγραμματιστικό περιβάλλον MakeCode



Εικόνα 11. Το προγραμματιστικό περιβάλλον MakeCode



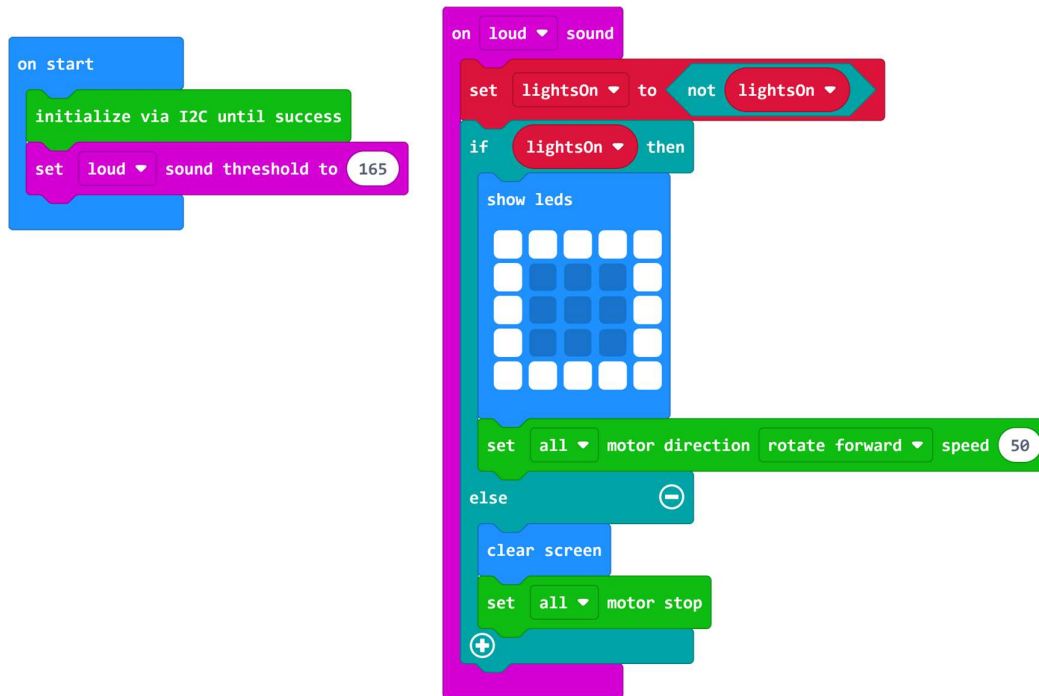
Εικόνα 12. Το προγραμματιστικό περιβάλλον MakeCode

4.2 Έργο 2: Κίνηση Maqueen Plus μέσω ήχου

Αυτό το έργο εστιάζει στη χρήση του Maqueen Plus για την ανάπτυξη προγράμματος που αξιοποιεί τον ήχο ως μέσο ελέγχου. Πριν τον κώδικα πρέπει να προετοιμάσουμε το Maqueen Plus V2.0.

Κώδικας και τεκμηρίωση

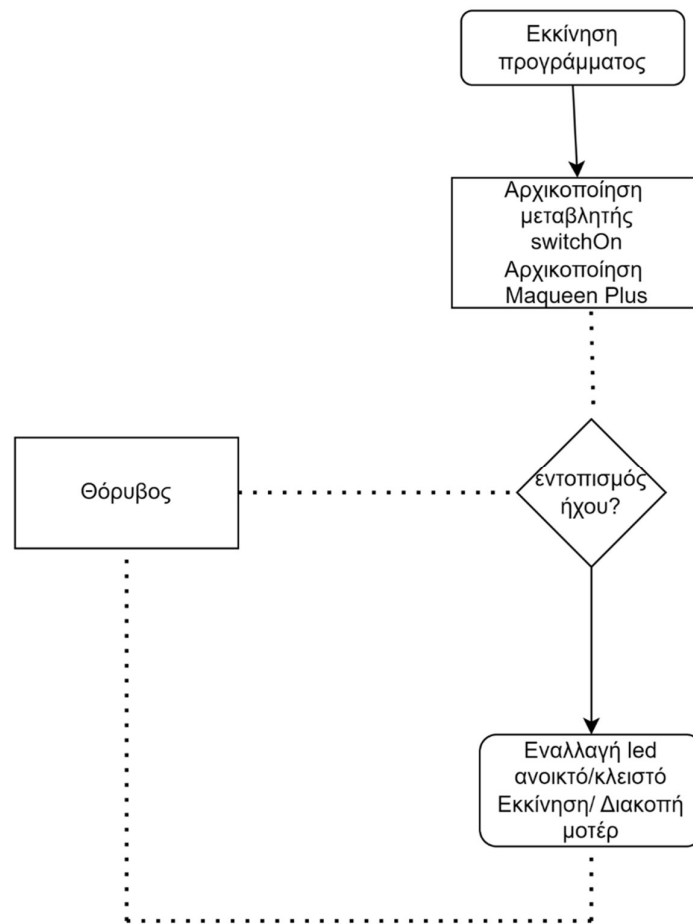
```
input.onSound(DetectedSound.Loud, function () {
  lightsOn = !(lightsOn)
  if (lightsOn) {
    basic.showLeds(`
      # # # # #
      # . . . #
      # . . . #
      # . . . #
      # # # # #
    `)
    maqueenPlusV2.controlMotor(maqueenPlusV2.MyEnumMotor.AllMotor, maqueenPlusV2.MyEnumDir.Forward, 50)
  } else {
    basic.clearScreen()
    maqueenPlusV2.controlMotorStop(maqueenPlusV2.MyEnumMotor.AllMotor)
  }
})
let lightsOn = false
maqueenPlusV2.I2CInit()
input.setSoundThreshold(SoundThreshold.Loud, 165)
```



Εικόνα 13. Κώδικας WordBlocks 'Κίνηση Maqueen Plus μέσω ήχου'

Τεκμηρίωση κώδικα:

- Ο κώδικας συνεχίζει στο έργο του προηγούμενο κεφαλαίου.
- Με την εκτέλεση του κώδικα, οι λαμπτήρες LED του micro:bit σβήνουν ή ανάβουν καθώς και τα μοτέρ του Maqueen Plus ξεκινούν ή σταματούν μέσω ενός δυνατού ήχου στο μικρόφωνο του micro:bit, την ευαισθησία του οποίου μπορούμε να ρυθμίσουμε από το κατώφλι ευαισθησίας.
- Η lights_on είναι μια μεταβλητή που χρησιμοποιείται για να παρακολουθεί την κατάσταση των LED και των μοτερ του Maqueen Plus.
- Η συνάρτηση on_sound_loud() εναλλάσσει την κατάσταση της lights_on και ανάβει ή σβήνει τα LED καθώς και ανοίγει ή κλείνει τα μοτερ του Maqueen Plus ανάλογα την τιμή της.
- Ο ακροατής συμβάντος (input.on_sound) ρυθμίζεται να καλεί την on_sound_loud όταν εντοπιστεί ένας δυνατός ήχος.
- Το κατώφλι ήχου για την ανίχνευση δυνατού ήχου ορίζεται στο 165 χρησιμοποιώντας το input.set_sound_threshold.
- Η γραμμή maqueenPlusV2.i2c_init() θέτει το Maqueen Plus στην αρχική του κατάσταση.



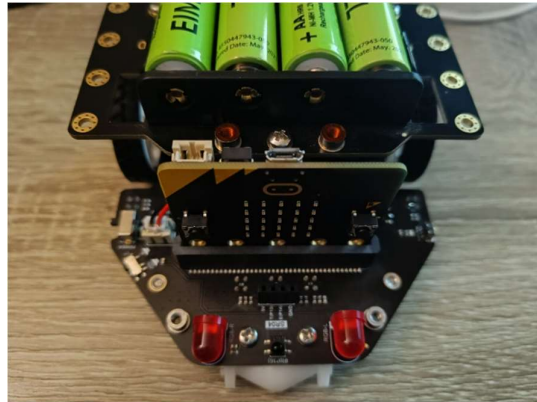
Εικόνα 14. Διάγραμμα ροής 'Κίνηση Maqueen Plus μέσω ήχου'.

Προετοιμασία και σύνδεση Maqueen Plus + micro:bit με τον υπολογιστή

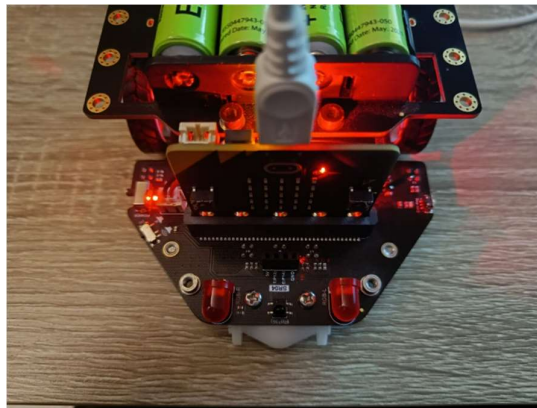
1. Τοποθετούμε 4 μπαταρίες (τύπου AA) στο Maqueen Plus
2. Στη συνέχεια, τοποθετούμε το micro:bit στην ειδική υποδοχή. [Εικόνα 15]
3. Συνδέουμε το micro:bit στον υπολογιστή χρησιμοποιώντας το καλώδιο USB. [Εικόνα 16]
4. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικά στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.
5. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.
6. Αφαιρούμε το καλώδιο USB και στην συνέχεια θέτουμε τον διακόπτη λειτουργίας του Maqueen Plus στην ενεργή του κατάσταση και το τοποθετούμε σε ένα επίπεδο στον χώρο. [Εικόνα 17]

7. Προσεγγίζουμε το Maqueen Plus (απόσταση περ. 30 εκατοστών) και κάνοντας ένα θόρυβο (π.χ. παλαμάκι) το όχημα θα ξεκινήσει.

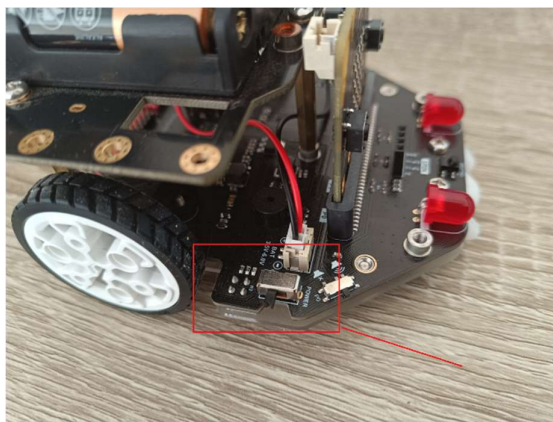
8. Επαναλαμβάνουμε το βήμα 6 και το όχημα θα σταματήσει.



Εικόνα 15. 'Κίνηση Maqueen Plus μέσω ήχου'.



Εικόνα 16. 'Κίνηση Maqueen Plus μέσω ήχου'.



Εικόνα 17. 'Κίνηση Maqueen Plus μέσω ήχου'.

Παρατηρήσεις

Η ευαισθησία του μικροφώνου του micro:bit εξαρτάται από την τιμή που θα ορίσουμε ως κατώφλι, οπότε με μικρότερη τιμή θα είναι ευκολότερο να ανάψουν/σβήσουν τα LEDs καθώς και να ξεκινήσει ή να σταματήσει το Maqueen Plus, όμως θα είναι επίσης ευκολότερο να συμβεί αυτό από λάθος, όπως π χ θόρυβος στο περιβάλλον ή ο θόρυβος που παράγεται από την κίνηση του οχήματος. Προσαρμόζουμε την τιμή στο κατώφλι, έως ότου έχουμε το επιθυμητό αποτέλεσμα.

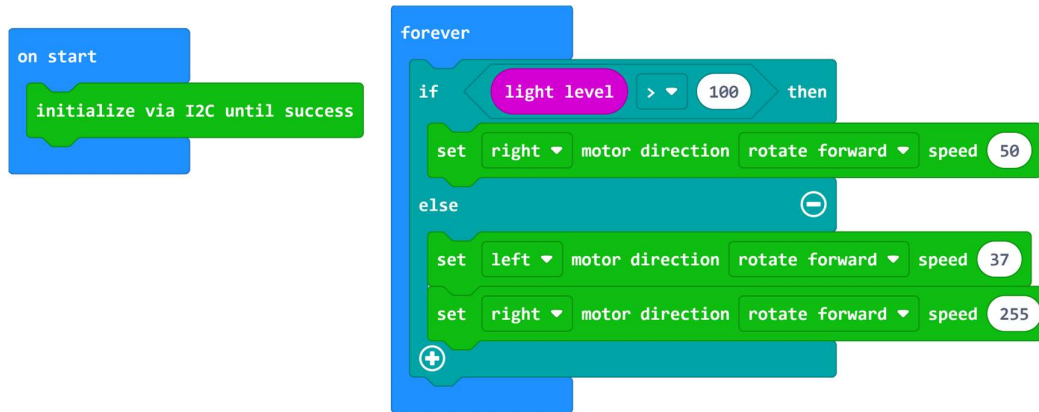
4.3 Έργο 3: Κίνηση Maqueen Plus μέσω φωτός

Σε αυτό το έργο θα δημιουργήσουμε ένα ρομπότ το οποίο είναι εμπνευσμένο από κάποια έντομα, τα οποία σε νυχτερινό περιβάλλον όταν βρίσκονται κοντά σε κάποια πηγή φωτός, κινούνται προς αυτό.

Κώδικας και τεκμηρίωση

```
maqueenPlusV2.i2c_init()

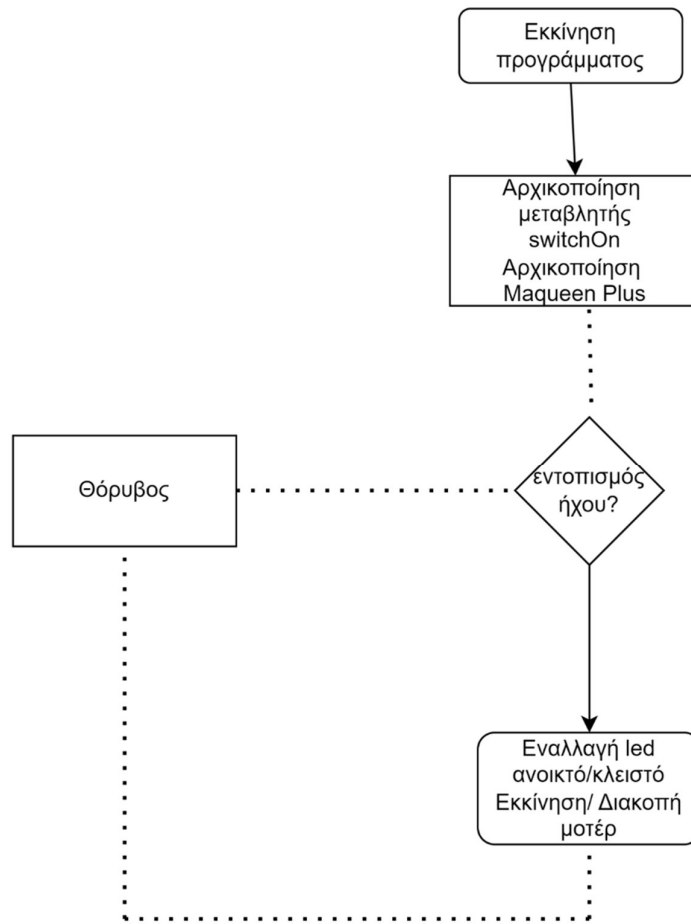
def on_forever():
    if input.light_level() > 100:
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.RIGHT_MOTOR,
                                     maqueenPlusV2.MyEnumDir.FORWARD,
                                     50)
    else:
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.LEFT_MOTOR,
                                     maqueenPlusV2.MyEnumDir.FORWARD,
                                     37)
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.RIGHT_MOTOR,
                                     maqueenPlusV2.MyEnumDir.FORWARD,
                                     255)
basic.forever(on_forever)
```



Εικόνα 18. Κώδικας WordBlocks 'Κίνηση Maqueen Plus μέσω φωτός'

Τεκμηρίωση κώδικα:

- `maqueenPlusV2.i2c_init()` Αρχικοποίηση του Maqueen Plus
- `basic.forever(on_forever)` Σε αυτή τη γραμμή η συνάρτηση `forever` θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση `on_forever`, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- Το Maqueen Plus διαθέτει αισθητήρα φωτός, την τιμή του οποίου μπορούμε να πάρουμε καλώντας την συνάρτηση `input.light_level()` την τιμή της οποίας συγκρίνουμε υπό συνθήκη με κάποιο σταθερό αριθμό της επιλογής μας (μικρότερη τιμή σημαίνει μεγαλύτερη ευαισθησία στο φως με μικρότερη τιμή 0 και μέγιστη τιμή 255)
- Στην πρώτη περίπτωση της συνθήκης, δηλαδή όταν εντοπίζεται φως δυνατότερο από το κατώφλι που έχουμε ορίσει, το Maqueen Plus αρχίζει να κινείται ευθεία προς την κατεύθυνση του φωτός
- Στην δεύτερη περίπτωση της συνθήκης, δεν έχει ανιχνευθεί επαρκές φως, οπότε το Maqueen Plus κινείται κυκλικά γύρω από τον εαυτό του. Αυτό το καταφέρνουμε, θέτοντας σημαντική διαφορά στις τιμές των μοτερ ώστε το ένα να περιστρέφεται σημαντικά πιο γρήγορα από το άλλο.
- Με αυτό το τρόπο το Maqueen Plus θα ανιχνεύει και θα κινείται προς τις πηγές φωτός που φέρνουμε κοντά του.



Εικόνα 19. Διάγραμμα ροής 'Κίνηση Maqueen Plus μέσω φωτός'.

Προετοιμασία και σύνδεση Maqueen Plus + micro:bit με τον υπολογιστή

1. Συνδέουμε το micro:bit στον υπολογιστή χρησιμοποιώντας το καλώδιο USB.
2. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικά στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.
3. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.
4. Αφαιρούμε το καλώδιο USB και στην συνέχεια θέτουμε τον διακόπτη λειτουργίας του Maqueen Plus στην ενεργή του κατάσταση και το τοποθετούμε σε ένα επίπεδο στον χώρο. Σε αυτό το σημείο το Maqueen Plus κινείται κυκλικά γύρω από τον εαυτό του.
5. Προσεγγίζουμε το Maqueen Plus με κάποιο φακό (απόσταση περίπου 30 εκατοστά ή λιγότερη).
6. Το Maqueen Plus θα αρχίσει να κινείται προς τον φακό.

7. Μετακινούμε μακριά τον φακό. Το Maqueen Plus κινείται και πάλι κυκλικά γύρω από τον εαυτό του.

Παρατηρήσεις

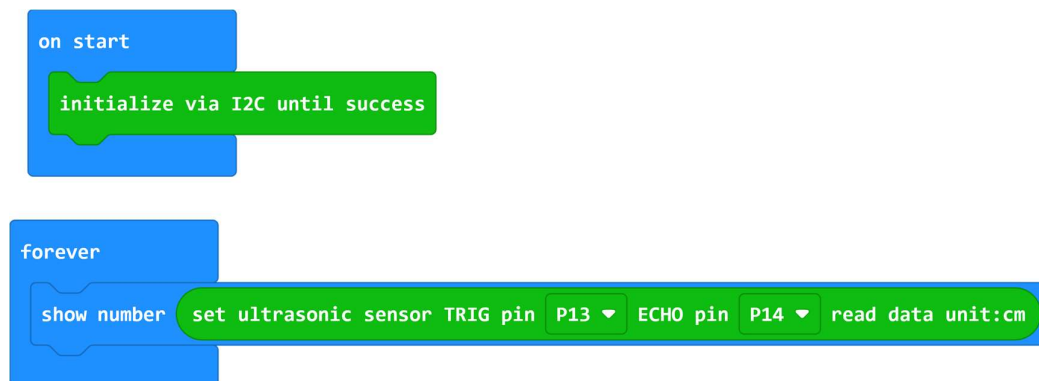
Η ευαισθησία του αισθητήρα φωτός του Maqueen Plus εξαρτάται από την τιμή που θα συγκρίνουμε στον κώδικα μας, οπότε με μικρότερη τιμή θα είναι ευκολότερο να ανάψουν/σβήσουν τα LEDs καθώς και να ξεκινήσει ή να σταματήσει το Maqueen Plus, όμως θα είναι επίσης ευκολότερο να συμβεί αυτό από λάθος, όπως π.χ. άλλες πηγές φωτός στο περιβάλλον. Προσαρμόζουμε την τιμή σύγκρισης, έως ότου έχουμε το επιθυμητό αποτέλεσμα.

4.4 Έργο 4: Μέτρηση απόστασης μέσω υπερήχων

Σε αυτό το έργο θα εξερευνήσουμε πώς το Maqueen Plus μπορεί να μετρήσει αποστάσεις χρησιμοποιώντας την συσκευή υπερήχων

Κώδικας και τεκμηρίωση

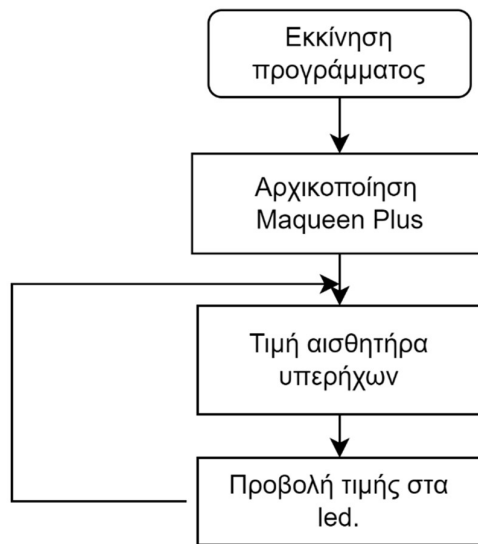
```
maqueenPlusV2.i2c_init()  
  
def on_forever():  
    basic.show_number(maqueenPlusV2.read_ultrasonic(DigitalPin.P13, DigitalPin.P14))  
basic.forever(on_forever)
```



Εικόνα 20. Κώδικας WordBlocks 'Μέτρηση απόστασης μέσω υπερήχων'

Τεκμηρίωση κώδικα:

- `maqueenPlusV2.i2c_init()` Αρχικοποίηση του Maqueen Plus
- `basic.forever(on_forever)` Σε αυτή τη γραμμή η συνάρτηση `forever` θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση `on_forever`, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- `maqueenPlusV2.read_ultrasonic(DigitalPin.P13, DigitalPin.P14)` αυτή η συνάρτηση παίρνει ως παραμέτρους τις υποδοχές του πομπού και του δέκτη. Ως πομπό χρησιμοποιούμε την είσοδο P13 και ως δέκτη την είσοδο P14, η συνάρτηση επιστρέφει αριθμό σε εκατοστά που υποδηλώνει την απόσταση από κάποιο αντικείμενο.
- `basic.show_number(maqueenPlusV2.read_ultrasonic(DigitalPin.P13, DigitalPin.P14))` Η `show_number` θα λάβει ως παράμετρο την τιμή που θα δώσει η συνάρτηση `read_ultrasonic(DigitalPin.P13, DigitalPin.P14)` και θα την προβάλλει στα LEDs του `micro:bit`.



Εικόνα 21. Διάγραμμα ροής 'Μέτρηση απόστασης μέσω υπερήχων'.

Προετοιμασία και σύνδεση Maqueen Plus + `micro:bit` με τον υπολογιστή

1. Τοποθετούμε το εξάρτημα υπερήχων στο Maqueen Plus όπως στην εικόνα 1. [Εικόνα 22]
2. Συνδέουμε το `micro:bit` στον υπολογιστή χρησιμοποιώντας το καλώδιο USB.

3. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικά στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.

4. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.

5. Αφαιρούμε το καλώδιο USB και στην συνέχεια θέτουμε τον διακόπτη λειτουργίας του

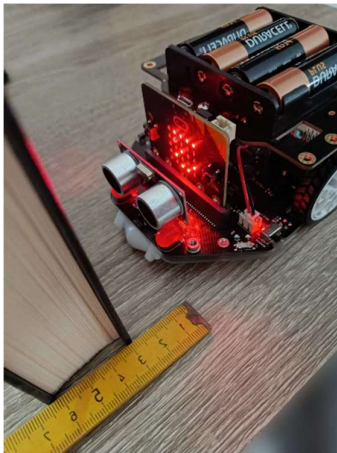
Maqueen Plus στην ενεργή του κατάσταση και το τοποθετούμε σε ένα επίπεδο στον χώρο.

6. Πλησιάζουμε κάποιο αντικείμενο (π.χ. βιβλίο) μπροστά από τον αισθητήρα υπερήχων. Παρατηρούμε ότι η τιμή αλλάζει με αρκετή ακρίβεια. [Εικόνα 23]

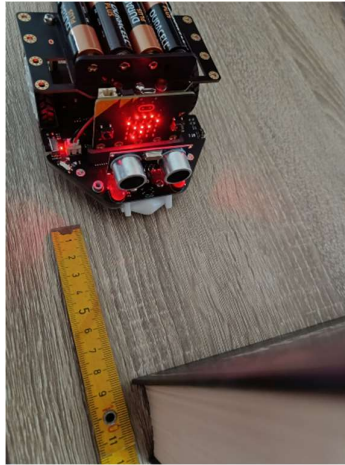
7. Μετακινούμε το αντικείμενο πιο μακριά από τον αισθητήρα υπερήχων. Παρατηρούμε ότι η τιμή αλλάζει και πάλι αρκετή ακρίβεια. [Εικόνα 24]



Εικόνα 22. Μέτρηση απόστασης μέσω υπερήχων.



Εικόνα 23. Μέτρηση απόστασης μέσω υπερήχων.



Εικόνα 24. Μέτρηση απόστασης μέσω υπερήχων.

Παρατηρήσεις

Η ακρίβεια στην μέτρηση του αισθητήρα υπερήχων μειώνεται όσο μεγαλώνει η απόσταση (π.χ. απόσταση μεγαλύτερη από 50 εκατοστά). Πολύ σημαντικό επίσης είναι οι μπαταρίες του Maqueen Plus να είναι πλήρως φορτισμένες, για να λειτουργεί σωστά ο αισθητήρας.

4.5 Έργο 5: Εντοπισμός εμποδίου και παύση κίνησης

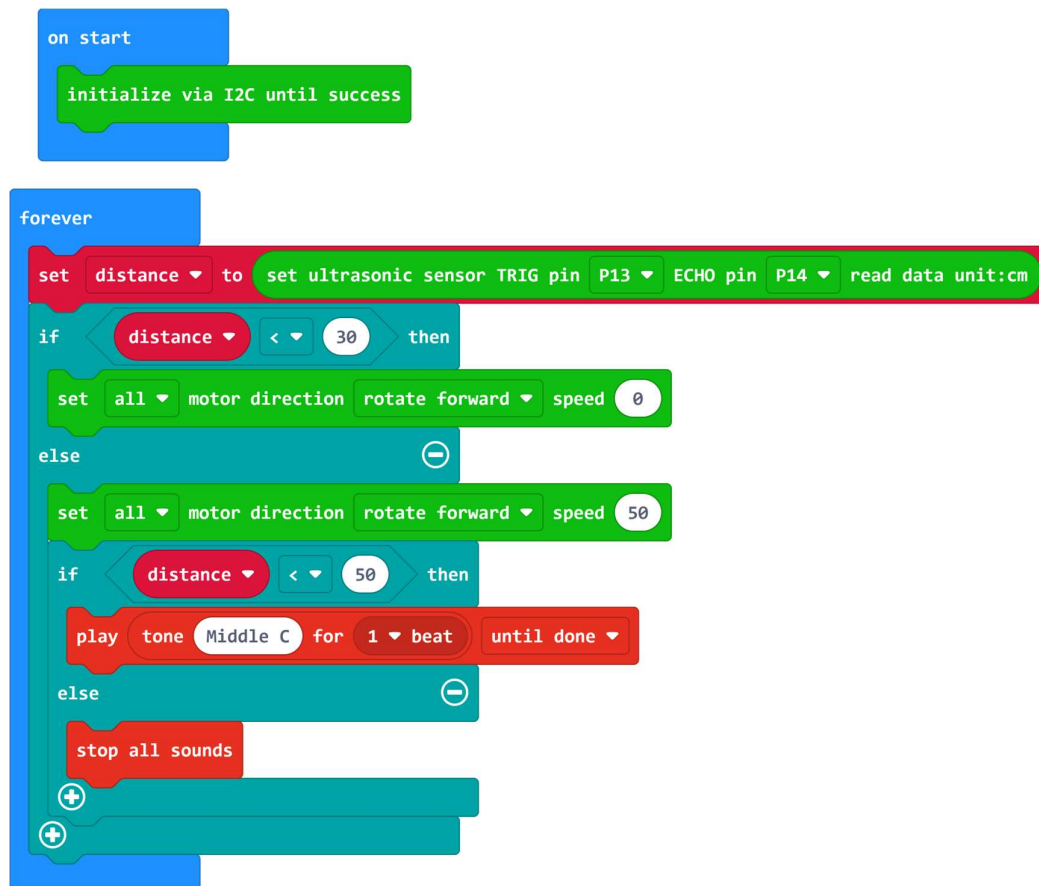
Εξελίσσοντας το προηγούμενο έργο, φτιάχνουμε ένα ρομπότ το οποίο σταματάει αυτόματα όταν βρεθεί μπροστά του εμπόδιο

Κώδικας και τεκμηρίωση

```
distance = 0
maqueenPlusV2.i2c_init()

def on_forever():
    global distance
    distance =
maqueenPlusV2.read_ultrasonic(DigitalPin.P13,
DigitalPin.P14)
    if distance < 30:
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumM
otor.ALL_MOTOR,
            maqueenPlusV2.MyEnumDir.FORWARD,
            0)
    else:
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumM
otor.ALL_MOTOR,
            maqueenPlusV2.MyEnumDir.FORWARD,
```

```
50)
    if distance < 50:
        music.play(music.tone_playable(262,
music.beat(BeatFraction.WHOLE)),
        music.PlaybackMode.UNTIL_DONE)
    else:
        music.stop_all_sounds()
basic.forever(on_forever)
```

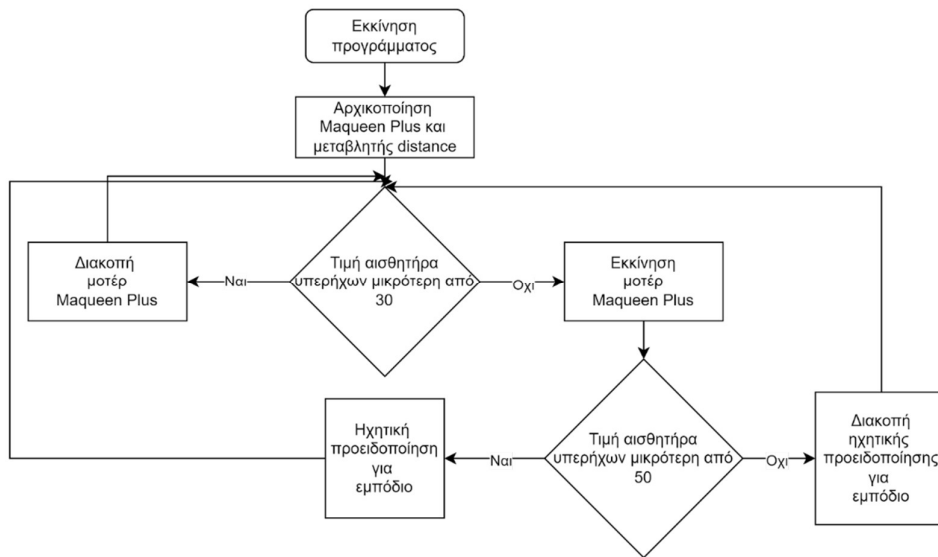


Εικόνα 25. Κώδικας WordBlocks 'Εντοπισμός εμποδίου και παύση κίνησης'

Τεκμηρίωση κώδικα:

- Η μεταβλητή distance θα κρατάει την τιμή που επιστρέφει ο αισθητήρας υπερήχων και την αρχικοποιούμε με 0.
- maqueenPlusV2.i2c_init() Αρχικοποίηση του Maqueen Plus

- `basic.forever(on_forever)` Σε αυτή τη γραμμή η συνάρτηση `forever` θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση `on_forever`, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- Μέσα στην συνθήκη `if`, ελέγχουμε αν η το αντικείμενο που βρίσκεται απέναντι από το Maqueen Plus είναι σε απόσταση μικρότερη των 30 εκατοστών. Σε αυτή την περίπτωση το Maqueen Plus θα σταματήσει να κινείται.
- Στην δεύτερη περίπτωση και τα δύο μοτέρ τίθενται σε λειτουργία και το Maqueen Plus κινείται ευθεία
- Στην δεύτερη συνθήκη συγκρίνουμε αν η απόσταση είναι μικρότερη των 50 εκατοστών και αν είναι αληθής, το Maqueen Plus παράγει ήχο για να μας ενημερώσει ότι προσεγγίζει εμπόδιο.
- Με αυτό το τρόπο το Maqueen Plus θα ανιχνεύει και θα σταματάει έγκαιρα αν βρει μπροστά του εμπόδιο.



Εικόνα 26. Διάγραμμα ροής 'Εντοπισμός εμπόδιου και παύση κίνησης'.

Προετοιμασία και σύνδεση Maqueen Plus + micro:bit με τον υπολογιστή

1. Συνδέουμε το micro:bit στον υπολογιστή χρησιμοποιώντας το καλώδιο USB
2. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικα στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.
3. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.

4. Αφαιρούμε το καλώδιο usb και στην συνέχεια θέτουμε τον διακόπτη λειτουργίας του Maqueen Plus στην ενεργή του κατάσταση και το τοποθετούμε σε ένα επίπεδο στον χώρο. Σε αυτό το σημείο το Maqueen Plus κινείται κυκλικά γύρω από τον εαυτό του.
5. Τοποθετούμε κάποιο εμπόδιο μπροστά στο Maqueen Plus περίπου ένα μέτρο. Το Maqueen Plus συνεχίζει να κινείται προς το εμπόδιο.
6. Μόλις ο αισθητήρας εντοπίσει το αντικείμενο σε απόσταση μικρότερη των 50 εκατοστών, ξεκινάει να ηχεί.
7. Σύντομα μόλις φτάσει σε απόσταση μικρότερη των 30 εκατοστών θα σταματήσει να κινείται.

4.6 Έργο 6: Line follower με τους αισθητήρες γραμμών

Το ρομπότ line follower έχει την δυνατότητα να κινείται πάνω σε έναν χάρτη που ορίζουμε εμείς. Η διαδρομή/γραμμή που ορίζουμε δεν έχει σημασία αν έχει στροφές ή είναι ευθεία, αρκεί να ορίσουμε σωστά τις παραμέτρους του Maqueen Plus και να φροντίσουμε οι αισθητήρες να μην υπερβαίνουν το πλάτος της γραμμής.

Κώδικας και τεκμηρίωση

```
maqueenPlusV2.i2c_init()

def on_forever():
    if
maqueenPlusV2.read_line_sensor_state(maqueenPlusV2.MyEnum
LineSensor.SENSOR_M) == 1:
        maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumM
otor.ALL_MOTOR,
            maqueenPlusV2.MyEnumDir.FORWARD,
            14)
        if
maqueenPlusV2.read_line_sensor_state(maqueenPlusV2.MyEnum
LineSensor.SENSOR_L1) == 1 and
maqueenPlusV2.read_line_sensor_state(maqueenPlusV2.MyEnum
LineSensor.SENSOR_R1) == 0:
            maqueenPlusV2.control_motor(maqueenPlusV2.MyE
numMotor.RIGHT_MOTOR,
                maqueenPlusV2.MyEnumDir.FORWARD,
                50)
            maqueenPlusV2.control_motor(maqueenPlusV2.MyE
numMotor.LEFT_MOTOR,
                maqueenPlusV2.MyEnumDir.FORWARD,
                2)
        if
maqueenPlusV2.read_line_sensor_state(maqueenPlusV2.MyEnum
LineSensor.SENSOR_L1) == 0 and
```



```
maqueenPlusV2.read_line_sensor_state(maqueenPlusV2.MyEnumLineSensor.SENSOR_R1) == 1:  
    maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.LEFT_MOTOR,  
                                maqueenPlusV2.MyEnumDir.FORWARD,  
                                50)  
    maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.RIGHT_MOTOR,  
                                maqueenPlusV2.MyEnumDir.FORWARD,  
                                2)  
else:  
    maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor.ALL_MOTOR,  
                                maqueenPlusV2.MyEnumDir.FORWARD,  
                                0)  
basic.forever(on_forever)
```



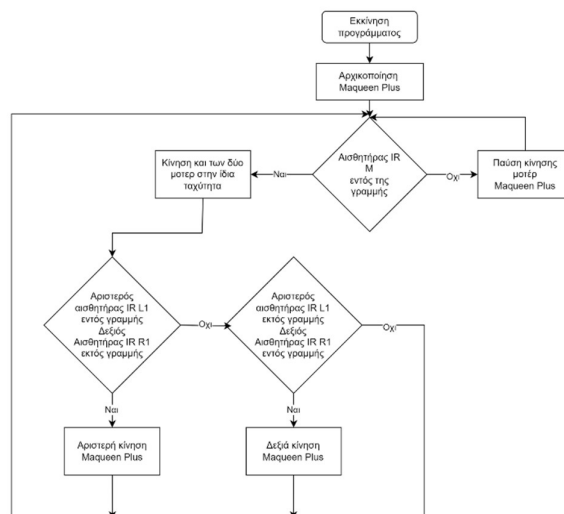
Εικόνα 27. Κώδικας WordBlocks 'Line follower με τους αισθητήρες γραμμών'

Τεκμηρίωση κώδικα:

- maqueenPlusV2.i2c_init() Αρχικοποίηση του Maqueen Plus
- basic.forever(on_forever) Σε αυτή τη γραμμή η συνάρτηση forever θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση on_forever, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- read_line_sensor_state(maqueenPlusV2.MyEnumLineSensor.SENSOR_L1) Αυτή η συνάρτηση, λαμβάνει σαν παράμετρο τον αισθητήρα που μας

ενδιαφέρει να ελέγξουμε την τιμή και στην συνέχεια επιστρέφει 1 ή 0 ανάλογα με το αν το όχημα βρίσκεται πάνω στην γραμμή ή όχι αντίστοιχα.

- Μέσα στην πρώτη συνθήκη if, γίνεται έλεγχος για το αν ο αισθητήρας υπερέθρων M βρίσκεται πάνω στην μαύρη γραμμή. Εάν συμβαίνει αυτό, τότε και τα δύο μοτέρ λειτουργούν στην ίδια ταχύτητα που έχουμε ορίσει και το Maqueen Plus κινείται ευθεία. Διαφορετικά τα μοτέρ σταματούν και το όχημα παύει να κινείται.
- Επίσης στην περίπτωση που το όχημα κινείται γίνονται περαιτέρω έλεγχοι. Αρχικά γίνεται έλεγχος αν ο αριστερός αισθητήρας γραμμών L1 ανιχνεύει την γραμμή ενώ ταυτόχρονα ο δεξιός R1 δεν την ανίχνευσε. Αυτό υποδηλώνει στο όχημα ότι βρίσκεται πάνω σε αριστερή στροφή και έχει ως αποτέλεσμα να δώσει πολύ μεγαλύτερη κίνηση στον δεξιά τροχό/μοτέρ ενώ παράλληλα πολύ μικρή κίνηση στον αριστερό τροχό/μοτέρ και το Maqueen Plus θα κινηθεί προς τα αριστερά
- Με παρόμοιο τρόπο, γίνεται έλεγχος αν ο αριστερός αισθητήρας γραμμών L1 δεν ανιχνεύει την γραμμή ενώ ταυτόχρονα ο δεξιός R1 την ανιχνεύει. Αυτό υποδηλώνει στο όχημα ότι βρίσκεται πάνω σε δεξιά στροφή και έχει ως αποτέλεσμα να δώσει πολύ μεγαλύτερη κίνηση στον αριστερό τροχό/μοτέρ ενώ παράλληλα πολύ μικρή κίνηση στον δεξιά τροχό/μοτέρ και το Maqueen Plus θα κινηθεί προς τα δεξιά.



Εικόνα 28. Διάγραμμα ροής 'Line follower με τους αισθητήρες γραμμών'.

Προετοιμασία και σύνδεση Maqueen Plus + micro:bit με τον υπολογιστή

1. Παρατηρούμε την κάτω πλευρά του Maqueen Plus, όπως στην εικόνα βλέπουμε τους τρεις αισθητήρες γραμμών L1, M, R1 που αντιστοιχούν στον αριστερό, μέσο, και δεξιό αισθητήρα αντίστοιχα. [Εικόνα 29]
2. Επειδή το μαύρο χρώμα της γραμμής απορροφά το φως (υπέρυθρο φως) που παράγει ο αισθητήρας κατά την μετάδοση, αυτό δεν αντανακλάται πίσω στον αισθητήρα λήψης και αυτό έχει ως αποτέλεσμα τιμή 1 στην έξοδο του.
3. Συνδέουμε το micro:bit στον υπολογιστή χρησιμοποιώντας το καλώδιο USB.
4. Στη συνέχεια, κάνουμε κλικ στο "Download" για να μεταφέρουμε τον κώδικά στο micro:bit. Θα ξεκινήσει ένας οδηγός ζεύξης του micro:bit με τον υπολογιστή.
5. Παρακολουθούμε έως ότου αναβοσβήνει το φως στο micro:bit, που θα υποδηλώσει την επιτυχή μεταφορά του κώδικα.
6. Αφαιρούμε το καλώδιο usb και στην συνέχεια θέτουμε τον διακόπτη λειτουργίας του Maqueen Plus στην ενεργή του κατάσταση και το τοποθετούμε πάνω σε ένα χάρτη line follower (μπορούμε να τυπώσουμε με την βοήθεια ενός εκτυπωτή).
7. Όσο και οι τρεις αισθητήρες βρίσκονται εντός της γραμμής το όχημα κινείται ευθεία
8. Αν ένας από τους αισθητήρες L1, R1 βρεθεί εκτός γραμμής το όχημα θα στρίψει δεξιά ή αριστερά αντίστοιχα
9. Στην περίπτωση που ο μέσος αισθητήρας βρεθεί εκτός γραμμής, το όχημα θα σταματήσει να κινείται.



Εικόνα 29. 'Line follower με τους αισθητήρες γραμμών'.

Παρατηρήσεις

Πολλοί παράμετροι μπορούν να επηρεάσουν την σωστή κίνηση του οχήματος πάνω στον χάρτη

- Αρχικά πρέπει να φροντίσουμε η γραμμή να είναι αρκετά μεγάλη ώστε να υπερκαλύπτει στο πλάτος και τους τρεις αισθητήρες.
- Επίσης το χρώμα με το οποίο είναι φτιαγμένη η γραμμή πρέπει να απορροφά καλά το φως, ορισμένοι μαρκαδόροι ή χρώματα εκτυπωτών δεν αποδίδουν καλά το μαύρο και ενδέχεται να υπάρχει ανάκλαση του υπέρυθρου φωτός στον αισθητήρα με αποτέλεσμα να μην λειτουργεί αποτελεσματικά.
- Τέλος η ταχύτητα των μοτέρ τόσο στην ευθεία αλλά και στις στροφές, έχει μεγάλη σημασία στο τελικό αποτέλεσμα αφού αν το όχημα κινείται πολύ γρήγορα υπάρχει πιθανότητα σε κάποια στροφή να μην γίνει έγκαιρη ανάγνωση της γραμμής και το όχημα να βγει εκτός πορείας.

5. Εφαρμογές Κάμερας HuskyLens

Σε αυτό το κεφάλαιο θα εξετάσουμε την ενσωμάτωση και τη χρήση της κάμερας HuskyLens στα έργα μας. Θα εξετάσουμε πώς μπορούμε να χρησιμοποιήσουμε αυτήν την τεχνολογία για την ανίχνευση αντικειμένων και την αλληλεπίδραση με το ρομπότ μας. Επιπλέον, θα εξετάσουμε πώς μπορούμε να ενσωματώσουμε αυτήν την τεχνολογία στα έργα μας και πώς μπορούμε να προγραμματίσουμε το ρομπότ μας για να αντιδρά σε διαφορετικά αντικείμενα και περιβάλλοντα με χρήση της κάμερας HuskyLens.

5.1 Έργο 7: Μουσική από χρώματα

Το έργο "Μουσική από χρώματα" συνδέει ήχους με χρώματα, σαν έναν τρόπο ανάγνωσης μουσικής παρτιτούρας. Η χρήση διαφορετικών χρωμάτων, τα οποία μπορεί να είναι χρωματιστές κάρτες, αντιπροσωπεύουν νότες που μπορεί να βρίσκεται πάνω

σε ένα πεντάγραμμο. Συνδέοντας ένα προκαθορισμένο χρώμα με μία προκαθορισμένη νότα, μπορούμε να παράγουμε τις 7 νότες της μουσικής.

Στο έργο αυτό χρησιμοποιείται η λειτουργία αναγνώρισης χρώματος της κάμερας HuskyLens για να αναγνωριστούν οι χρωματιστές κάρτες. Η κίνηση του Maqueen Plus κατά την διάρκεια αναγνώρισης χρωμάτων από την κάμερα αντιπροσωπεύει τον ρυθμό και το τέμπο της μουσικής. Τοποθετώντας με διαφορετική σειρά τις κάρτες, δημιουργούμε διαφορετικές μελωδίες.

Προετοιμασία κάμερας / εκμάθηση χρωμάτων

1. Συνδέουμε την κάμερα Huskylens με ένα καλώδιο usb. Για να περιηγηθείτε στο μενού, στρέφουμε την ροδέλα δεξιά ή αριστερά έως ότου φέρουμε την επιλογή Αναγνώρισης χρώματος (Color Recognition). Στην συνέχεια πατάμε και κρατάμε παρατεταμένα την ροδέλα ώστε να επιβεβαιώσουμε την επιλογή μας. *[Εικόνα 30]*

2. Τώρα, θα πρέπει να βρισκόμαστε στο υπομένου της αναγνώρισης χρωμάτων. Πηγαίνουμε στην επιλογή εκμάθησης πολλαπλών(Learn Multiple) και πατάμε μία φορά την ροδέλα προς τα μέσα, για να αλλάξουμε την ρύθμιση. *[Εικόνα 31]*

3. Με την ροδέλα, φροντίζουμε να έχουμε το πλήκτρο της εικόνας 3 στην δεξιά πλευρά και ξαναπάμε μία φορά την ροδέλα προς τα μέσα για επιβεβαίωση.

4. Στρέφουμε και έπειτα πατάμε την ροδέλα στην επιλογή Save & return και επιλέγουμε και πατάμε το Yes για να ολοκληρώσουμε και να αποθηκεύσουμε τις ρυθμίσεις. *[Εικόνα 32]*

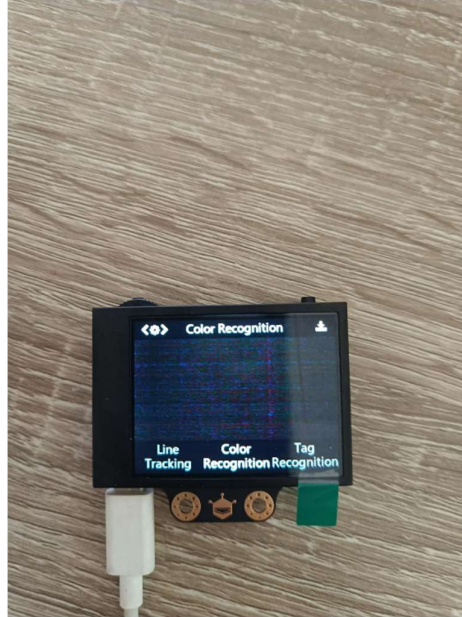
5. Για την εκμάθηση των χρωμάτων, φέρνουμε την κάμερα πάνω από κάποιο σχήμα χρώματος που θέλουμε να κάνουμε εκμάθηση (π.χ. χρωματιστά τετράγωνα της επιλογής μας με μαρκαδόρους σε απλό χαρτί, ή απλές χρωματιστές κάρτες κ.τ.λ.). Φροντίζουμε το πλαίσιο εκμάθησης που σχηματίζεται στην οθόνη, να καλύπτει όσο το δυνατόν περισσότερο το σχήμα που έχουμε σχεδιάσει. *[Εικόνα 33]*

6. Με το πλαίσιο πάνω από το χρώμα, πατάμε και κρατάμε παρατεταμένα (1-2 δευτερόλεπτα) το πλήκτρο εκμάθησης στο δεξί πάνω μέρος της κάμερας. Θα παρατηρήσουμε την ένδειξη εκμάθησης (Learning). *[Εικόνα 34]*

7. Σε αυτό το σημείο έχει γίνει εκμάθηση του πρώτου χρώματος, στο οποίο έχει αποδοθεί το αναγνωριστικό νούμερο 1 (Color ID: 1). Αφήνουμε το πλήκτρο εκμάθησης και μας εμφανίζεται ένα παράθυρο που μας ρωτάει αν θέλουμε να συνεχίσουμε με την

εκμάθηση των υπολοίπων χρωμάτων. Πάμε ξανά το πλήκτρο εκμάθησης(χρόνος επιβεβαίωσης 5 δευτερόλεπτα) ώστε να συνεχίσουμε. [Εικόνα 35]

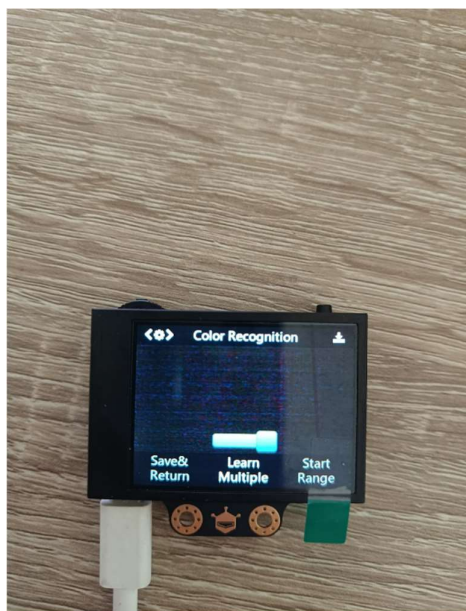
8. Επαναλαμβάνουμε τα βήματα 6 έως 7, ώστε να μάθουμε και τα 7 διαφορετικά χρώματα [Εικόνα 36]



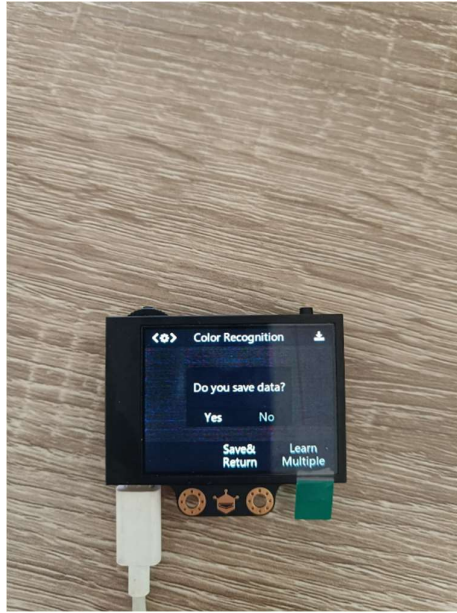
Εικόνα 30. 'Μουσική από χρώματα'.



Εικόνα 31. 'Μουσική από χρώματα'.



Εικόνα 32. 'Μουσική από χρώματα'.



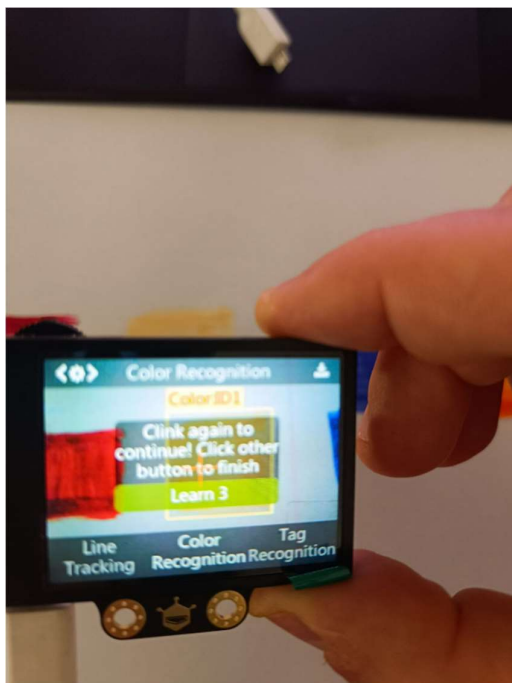
Εικόνα 33. 'Μουσική από χρώματα'.



Εικόνα 34. 'Μουσική από χρώματα'.



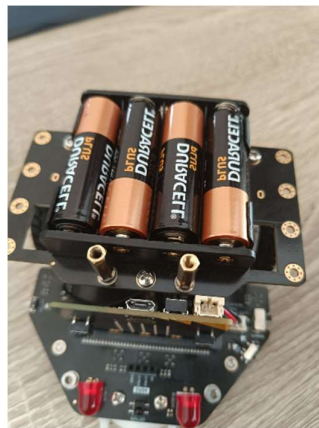
Εικόνα 35. 'Μουσική από χρώματα'.



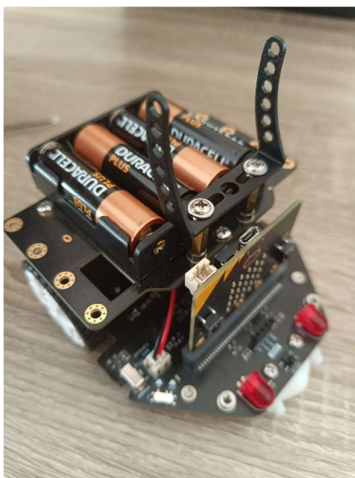
Εικόνα 36. 'Μουσική από χρώματα'.

Εγκατάσταση της κάμερας HuskyLens στο Maqueen Plus

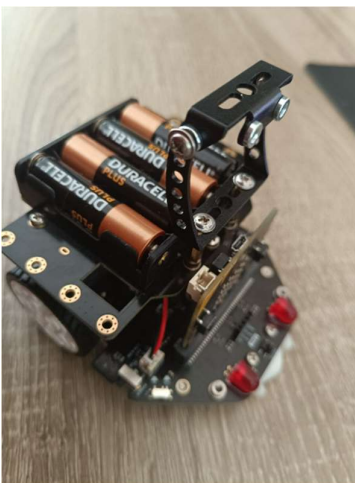
1. Τοποθετούμε τις χάλκινες εξάγωνες βίδες στις υποδοχές που βρίσκονται μπροστά από τις μπαταρίες. *[Εικόνα 37]*
2. Τοποθετούμε και με δύο βίδες βιδώνουμε το καμπυλωτό υποστήριγμα πάνω στις χάλκινες βίδες, με την κοίλη πλευρά να ‘κοιτάει’ προς την ίδια πλευρά με το Maqueen Plus *[Εικόνα 38]*
3. Τοποθετούμε και βιδώνουμε το δεύτερο υποστήριγμα με τις υποδοχές της κάμερας προς τα εμπρός. *[Εικόνα 39]*
4. Τοποθετούμε και βιδώνουμε το HuskyLens στις δύο υποδοχές του στηρίγματος του βήματος 3 με την κάμερα να κοιτάζει προς τα εμπρός. *[Εικόνα 40]*
5. Εφαρμόζουμε το καλώδιο σύνδεσης στον υποδοχέα του HuskyLens *[Εικόνα 41]*
6. Περνάμε την άλλη πλευρά του καλωδίου από την εσοχή, που βρίσκεται πάνω από τον δεξιό τροχό του Maqueen Plus. *[Εικόνα 42]*
7. Με ιδιαίτερη προσοχή στην σωστή συνδεσμολογία τοποθετούμε τα άκρα του καλωδίου στην υποδοχή I2C του Maqueen Plus, που βρίσκεται χαμηλά στην πίσω αριστερή πλευρά του οχήματος (δίπλα από τον αριστερό τροχό). *[Εικόνα 43]*
 - Πράσινο καλώδιο στο πράσινο pin SDA
 - Μπλε καλώδιο στο μπλε pin SCL
 - Μαύρο καλώδιο στο μαύρο pin GND
 - Κόκκινο καλώδιο στο κόκκινο pin 5V
8. Με τα προηγούμενα βήματα ολοκληρώνουμε την εγκατάσταση της κάμερας HuskyLens στο Maqueen Plus. Το όχημα είναι πλέον έτοιμο για να το προγραμματίσουμε. *[Εικόνα 44]*



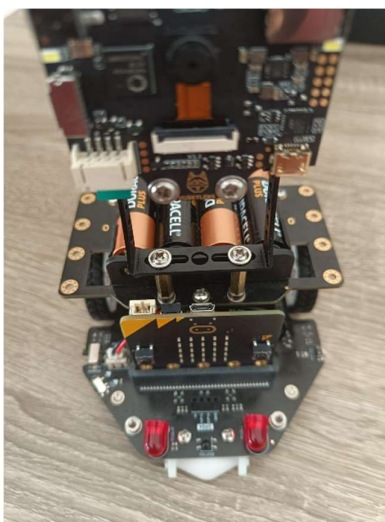
Εικόνα 37. ‘Μουσική από χρώματα’.



Εικόνα 38. 'Μουσική από χρώματα'.



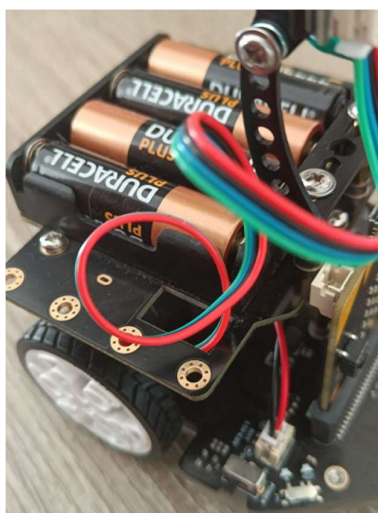
Εικόνα 39. 'Μουσική από χρώματα'.



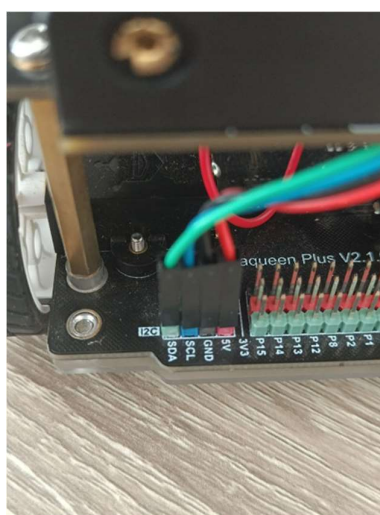
Εικόνα 40. 'Μουσική από χρώματα'.



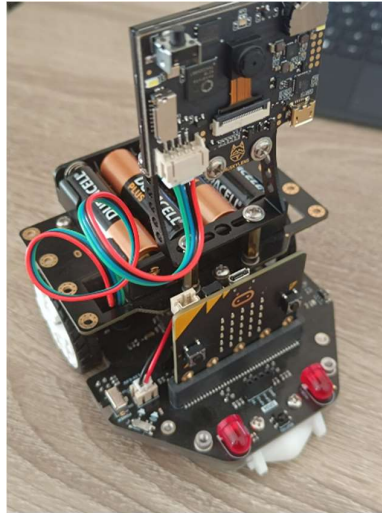
Εικόνα 41. 'Μουσική από χρώματα'.



Εικόνα 42. 'Μουσική από χρώματα'.



Εικόνα 43. 'Μουσική από χρώματα'.

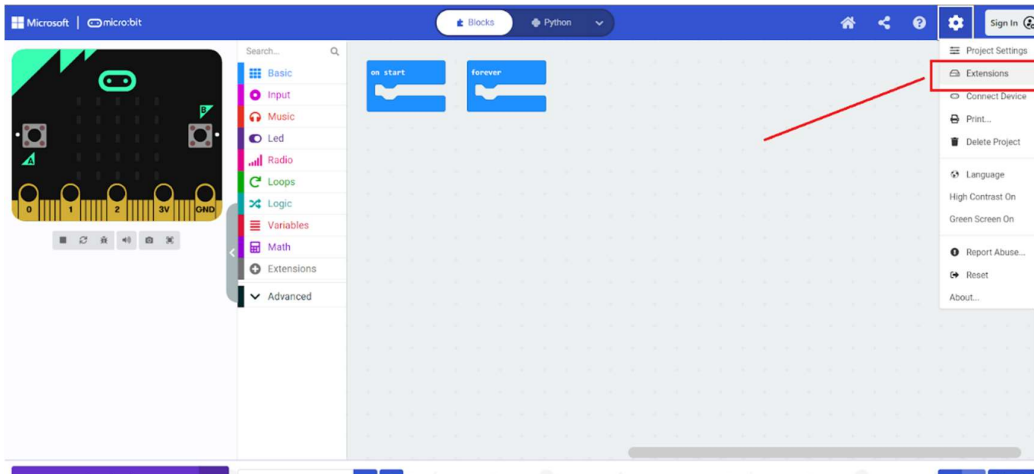


Εικόνα 44. 'Μουσική από χρώματα'.

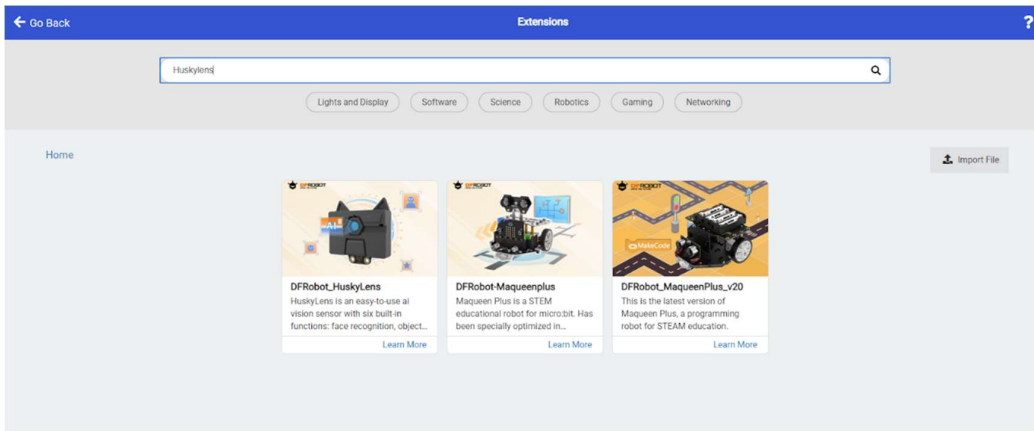
Κατανόηση του περιβάλλοντος MakeCode για προγραμματισμό Maqueen Plus + HuskyLens

1. Ανοίγουμε τον web browser και πλοηγούμαστε στη σελίδα <https://makecode.micro:bit.org/>.
2. Κάνουμε κλικ σε "New Project" για να φτιάξουμε ένα καινούργιο έργο και ορίζουμε ένα όνομα της επιλογής μας και το ανοίγουμε.
3. Σε αυτό το σημείο θα χρειαστεί να εισάγουμε την βιβλιοθήκη που είναι υπεύθυνη για το HuskyLens. Φροντίζουμε να έχουμε ήδη εγκατεστημένη την βιβλιοθήκη για το Maqueen Plus, όπως στις οδηγίες του προηγούμενου κεφαλαίου. Ανοίγουμε τις ρυθμίσεις(γρανάζι) και πατάμε στην επιλογή "Extensions" [Εικόνα 45]
4. Στην συνέχεια γράφουμε στο πεδίο αναζήτησης "HuskyLens" και πατώντας enter θα μας εμφανίσει το "DFRobot_HuskyLens" το οποίο και επιλέγουμε και αυτομάτως θα ακολουθήσει η εγκατάσταση της βιβλιοθήκης. [Εικόνα 46]
5. Σε αυτό το σημείο η εγκατάσταση της βιβλιοθήκης θα πρέπει να έχει ολοκληρωθεί επιτυχώς και θα βλέπουμε την επιλογή HuskyLens (καθώς επίσης θα πρέπει να έχουμε και το maqueenPlusV2) στην εργαλειοθήκη του MakeCode. [Εικόνα 47]

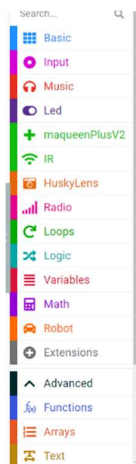
ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΜΕ WORDBLOCKS ΚΑΙ ΤΗΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON, ΜΕ ΤΗΝ AI CAMERA HUSKYLENS ΚΑΙ ΤΟ MICRO:BIT



Εικόνα 45. 'Μουσική από χρώματα'.



Εικόνα 46. 'Μουσική από χρώματα'.



Εικόνα 47. 'Μουσική από χρώματα'.

Κώδικας και τεκμηρίωση

```
ID = 0
maqueenPlusV2.i2c_init()
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_COLOR_RECOGNITION)

def on_forever():
    global ID
    huskylens.request()
    ID = huskylens.readBox_s(Content3.ID)
    maqueenPlusV2.control_motor(maqueenPlusV2.MyEnumMotor
.ALL_MOTOR,
    maqueenPlusV2.MyEnumDir.FORWARD,
    20)
    if ID == 1:
        music.play(music.tone_playable(262,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 2:
        music.play(music.tone_playable(294,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 3:
        music.play(music.tone_playable(330,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 4:
        music.play(music.tone_playable(349,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 5:
        music.play(music.tone_playable(392,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 6:
        music.play(music.tone_playable(440,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
    elif ID == 7:
        music.play(music.tone_playable(494,
music.beat(BeatFraction.DOUBLE)),
            music.PlaybackMode.UNTIL_DONE)
basic.forever(on_forever)
```

```
on start
  initialize via I2C until success
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Color Recognition

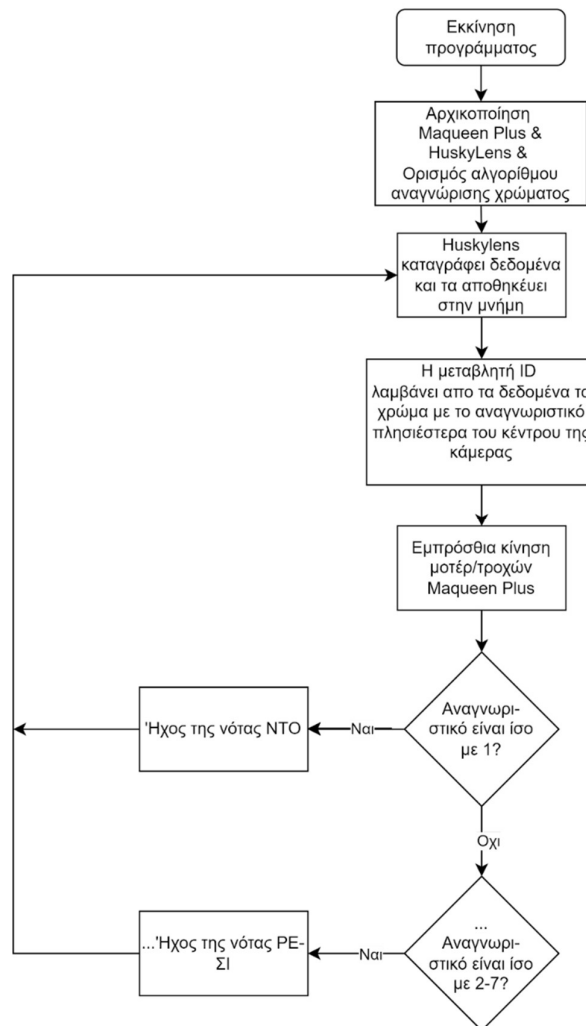
forever
  HuskyLens request data once and save into the result
  set ID to HuskyLens get ID of frame closest to the center of screen from the result
  set all motor direction rotate forward speed 20
  if ID = 1 then
    play tone Middle C for 2 beat until done
  else if ID = 2 then
    play tone Middle D for 2 beat until done
  else if ID = 3 then
    play tone Middle E for 2 beat until done
  else if ID = 4 then
    play tone Middle F for 2 beat until done
  else if ID = 5 then
    play tone Middle G for 2 beat until done
  else if ID = 6 then
    play tone Middle A for 2 beat until done
  else if ID = 7 then
    play tone Middle B for 2 beat until done
```

Εικόνα 48. Κώδικας WordBlocks 'Μουσική από χρώματα'

Τεκμηρίωση κώδικα:

- Δημιουργία μεταβλητής ID η οποία θα κρατάει την τιμή του κάθε χρώματος με την σειρά με την οποία εκπαιδεύσετε την κάμερα
- maqueenPlusV2.i2c_init() Αρχικοποίηση του Maqueen Plus
- huskylens.init_i2c() Αρχικοποίηση του HuskyLens
- huskylens.init_mode(protocolAlgorithm.ALGORITHM_COLOR_RECOGNITION) Με αυτή την γραμμή λέμε στο HuskyLens να χρησιμοποιήσει τον αλγόριθμο αναγνώρισης χρώματος
- basic.forever(on_forever) Σε αυτή τη γραμμή η συνάρτηση forever θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση on_forever, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- huskylens.request() Σε αυτή τη γραμμή λέμε το HuskyLens να κρατήσει τα δεδομένα που συλλέγει την συγκεκριμένη στιγμή στην μνήμη του.
- ID = huskylens.readBox_s(Content3.ID) Σε αυτή τη γραμμή ζητάμε από το HuskyLens να μας δώσει την αναγνωριστική τιμή του χρώματος που έχει εκπαιδευτεί και βρίσκεται το δυνατόν πλησιέστερα στο κέντρο της οθόνης, στην συνέχεια, ορίζουμε αυτή την τιμή στην μεταβλητή ID που φτιάξαμε
- maqueenPlusV2.control_motor(...) Ορίζουμε μία χαμηλή ταχύτητα κίνησης για τα μοτέρ/τροχούς του Maqueen Plus

- Στην συνέχεια έχουμε διαδοχικά συνθήκες, όπου συγκρίνουμε την τιμή ID με προκαθορισμένες τιμές που έχουμε ορίσει από το 1 έως το 7, που αντιστοιχούν στις νότες.
- Για παράδειγμα, αν η τιμή είναι ίση με 1, θα εκτελεστεί η πρώτη περίπτωση
- `music.tone_playable(262, music.beat(BeatFraction.DOUBLE))`, `music.PlaybackMode.UNTIL_DONE`) Ορίζουμε έναν ήχο διπλής διάρκειας ο οποίος θα παιχτεί έως το τέλος του προγράμματος και είναι 262 Hz που αντιστοιχεί στην νότα Ντο.
- `music.play()` Η συνάρτηση αυτή θα αναπαράγει τον ήχο με όλες του τις ρυθμίσεις που ορίσαμε ως παράμετρο
- Με όμοιο τρόπο γράφουμε και τις υπόλοιπες περιπτώσεις που αποτελούν και τις υπόλοιπες νότες (Ρε, Μι, Φα, Σολ, Λα, Σι)



Εικόνα 49. Διάγραμμα ροής 'Μουσική από χρώματα'.

Παρατηρήσεις

- Για την σωστή αναγνώριση των χρωμάτων, απαιτείται να έχουμε επαρκές φωτισμό στον χώρο ούτως ώστε τα χρώματα να φαίνονται καλά. Επίσης το επίπεδο φωτισμού θα πρέπει να μην αλλάζει από την προετοιμασία της κάμερας έως την τελική λειτουργία/δοκιμή του έργου. Ορισμένα χρώματα λειτουργούν καλύτερα από κάποια άλλα, όπως για παράδειγμα το πορτοκαλί μπορεί εύκολα να προσδιοριστεί λανθασμένα ως κόκκινο. Εδώ, αντιλαμβανόμαστε και τα όρια των δυνατοτήτων που έχει η κάμερα.
- Δοκιμάζουμε διαφορετικά χρώματα και επιλέγουμε αυτά που αναγνωρίζονται ευκολότερα. Επίσης πρέπει να φροντίσουμε η κάμερα να έχει επαρκή κλίση προς το έδαφος και να μην ‘κοιτάει’ ευθεία μπροστά, ώστε να αναγνωρίζει το δυνατόν ένα χρωματισμένο σχήμα την φορά στο έδαφος.
- Οι αποστάσεις που χωρίζουμε τα χρώματα, καθώς επίσης και παράμετροι, όπως η διάρκεια που επιλέγουμε σε κάθε νότα και η ταχύτητα των τροχών, δίνουν διαφορετικό αποτέλεσμα ανάλογα το τι θέλουμε να πετύχουμε.

5.2 Έργο 8: Line follower μέσω της κάμερας HuskyLens

Υπάρχουν αρκετές μέθοδοι με τις οποίες μπορούμε να επιτύχουμε ένα ρομπότ line follower. Στο έργο 6, χρησιμοποιούμε τους αισθητήρες γραμμών του Maqueen Plus για να το πετύχουμε. Σε αυτό το έργο αξιοποιούμε την δυνατότητα αναγνώρισης γραμμών που μας παρέχει η κάμερα HuskyLens. Υπάρχουν πλεονεκτήματα και μειονεκτήματα στις δυο αυτές μεθόδους. Κάποια από αυτά μπορούμε να δούμε στον παρακάτω πίνακα:

	Αισθητήρας γραμμών	Αναγνώριση γραμμών Huskylens
Οπτικό εύρος	Ο αισθητήρας πρέπει να βρίσκεται πολύ κοντά στο έδαφος. Περιορισμένο οπτικό εύρος	Η κάμερα μπορεί να έχει πολύ μεγαλύτερο οπτικό εύρος. Καλύτερη προσαρμογή σε διαφορετικούς χάρτες γραμμών
Προσαρμογή σε συνθήκες	Όταν αλλάζουν οι συνθήκες (φωτισμός, μέγεθος γραμμών, κτλ) ο αισθητήρας πρέπει να επαναρυθμίζεται.	Όταν αλλάζουν οι συνθήκες, αρκεί να ξαναγίνει εκμάθηση γραμμών.
Προσαρμογή σε χάρτη	Ο αισθητήρας αναγνωρίζει μόνο διαβαθμίσεις του γκρι. Οπότε μαύρες ή άσπρες γραμμές σε άσπρο ή μαύρο φόντο αντίστοιχα λειτουργούν αποτελεσματικά	Η αναγνώριση γραμμών μπορεί να διακρίνει και γραμμές διαφορετικών χρωμάτων

Πίνακας 2. Συνοπτική σύγκριση μεθόδων αναγνώρισης γραμμών της κάμερα HuskyLens και των αισθητήρων γραμμών του Maqueen Plus

Προετοιμασία κάμερας / εκμάθηση χρωμάτων

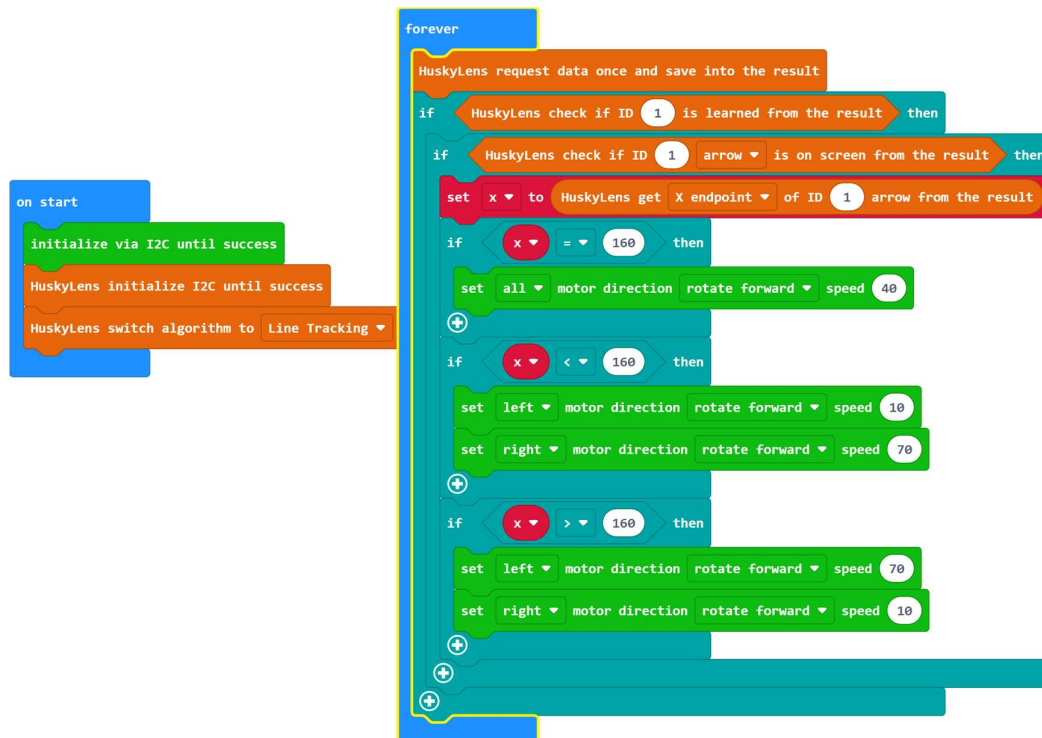
1. Συνδέουμε την κάμερα Huskylens με ένα καλώδιο usb. Για να περιηγηθούμε στο μενού, στρέφουμε την ροδέλα δεξιά ή αριστερά έως ότου φέρουμε την επιλογή Αναγνώρισης γραμμών(Line Tracking). Στην συνέχεια πατάμε και κρατάμε παρατεταμένα την ροδέλα ώστε να επιβεβαιώσουμε την επιλογή μας.
2. Τώρα, θα πρέπει να βρισκόμαστε στο υπομενού της αναγνώρισης γραμμών. Πηγαίνουμε στην επιλογή εκμάθησης πολλαπλών(Learn Multiple) και πατάμε μία φορά την ροδέλα προς τα μέσα, για να αλλάξουμε την ρύθμιση.
3. Με την ροδέλα, φροντίζουμε να έχουμε το πλήκτρο στην αριστερή πλευρά (θα κάνουμε εκμάθηση μόνο μίας γραμμής) και ξαναπάμε μία φορά την ροδέλα προς τα μέσα για επιβεβαίωση.
4. Στρέφουμε και έπειτα πατάμε την ροδέλα στην επιλογή Save & return και επιλέγουμε και πατάμε το Yes για να ολοκληρώσουμε και να αποθηκεύσουμε τις ρυθμίσεις.
5. Για την εκμάθηση της γραμμής, φέρνουμε την κάμερα πάνω από την γραμμή στον χάρτη. Φροντίζουμε η γραμμή που σχηματίζεται στην οθόνη, να είναι παράλληλη της κλήσης της γραμμής στον χάρτη. Πατάμε το πλήκτρο εκμάθησης, η γραμμή θα γίνει μπλε, που θα μας υποδηλώσει επιτυχή εκμάθηση.

Κώδικας και τεκμηρίωση

```
x = 0
maqueenPlusV2.i2c_init()
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_LINE_TRACKING)

def on_forever():
    global x
    huskylens.request()
    if huskylens.is_learned(1):
        if huskylens.is_appear(1,
HUSKYLENSResultType_t.HUSKYLENS_RESULT_ARROW):
            x = huskylens.read_arrow(1,
Content2.X_TARGET)
            if x == 160:
                maqueenPlusV2.control_motor(maqueenPlusV2
.MyEnumMotor.ALL_MOTOR,
                maqueenPlusV2.MyEnumDir.FORWARD,
                40)
            if x < 160:
                maqueenPlusV2.control_motor(maqueenPlusV2
.MyEnumMotor.LEFT_MOTOR,
                maqueenPlusV2.MyEnumDir.FORWARD,
                10)
                maqueenPlusV2.control_motor(maqueenPlusV2
.MyEnumMotor.RIGHT_MOTOR,
                maqueenPlusV2.MyEnumDir.FORWARD,
```

```
        70)
    if x > 160:
        maqueenPlusV2.control_motor (maqueenPlusV2
        .MyEnumMotor.LEFT_MOTOR,
        maqueenPlusV2.MyEnumDir.FORWARD,
        70)
        maqueenPlusV2.control_motor (maqueenPlusV2
        .MyEnumMotor.RIGHT_MOTOR,
        maqueenPlusV2.MyEnumDir.FORWARD,
        10)
basic.forever (on_forever)
```

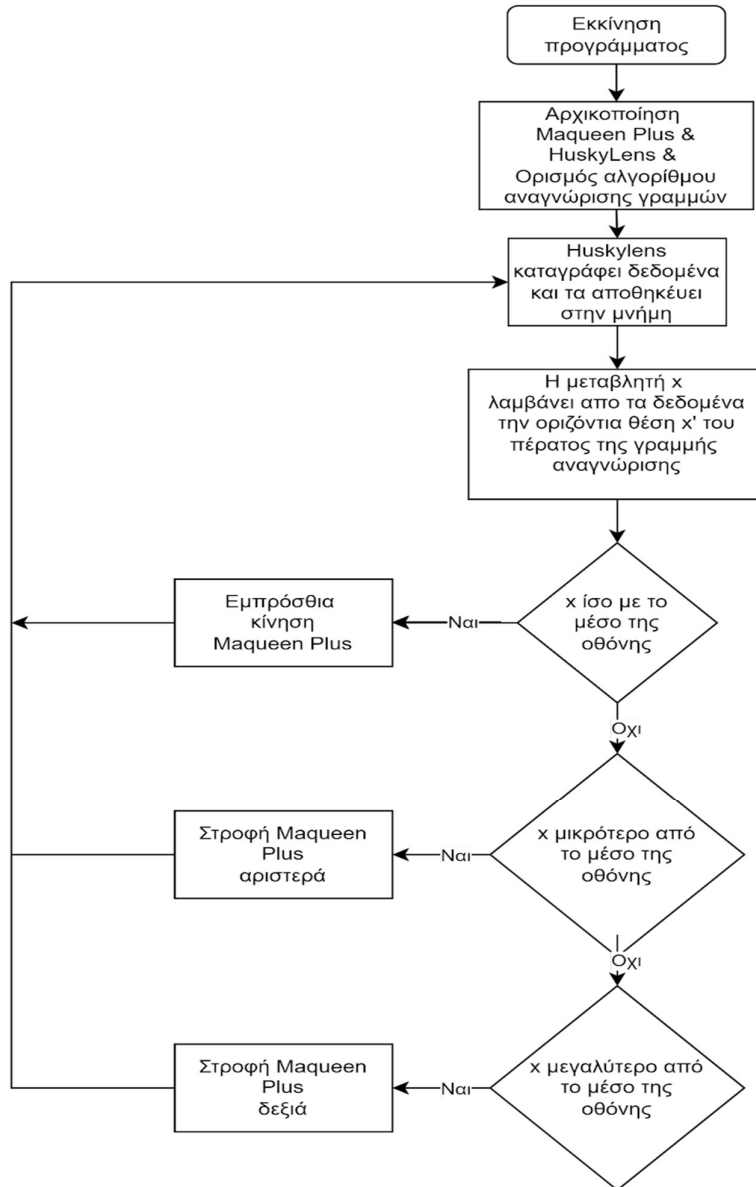


Εικόνα 50. Κώδικας WordBlocks 'Line follower μέσω της κάμερας HuskyLens'

Τεκμηρίωση κώδικα:

- maqueenPlusV2.i2c_init() Αρχικοποίηση του Maqueen Plus
- huskylens.init_i2c() Αρχικοποίηση του HuskyLens

- `huskylens.init_mode(protocolAlgorithm.ALGORITHM_LINE_TRACKING)`
Με αυτή την γραμμή λέμε στο HuskyLens να χρησιμοποιήσει τον αλγόριθμο αναγνώρισης γραμμής
- `basic.forever(on_forever)` Σε αυτή τη γραμμή η συνάρτηση `forever` θα εκτελεί διαρκώς τον κώδικα που βρίσκεται μέσα στην συνάρτηση `on_forever`, έως ότου κλείσουμε τον διακόπτη του Maqueen Plus
- Δηλώνουμε την καθολική μεταβλητή x . Σκοπός της μεταβλητής x είναι να κρατάει την τιμή x του πέρατος του διανύσματος που παράγεται κατά την αναγνώριση της γραμμής στον χάρτη. Η τιμή αυτή πρόκειται για την οριζόντια θέση του εικονοστοιχείου (pixel) της οθόνης, πάνω στο οποίο δείχνει το πέρασ του διανύσματος. Η οθόνη του Huskylens είναι 320X240 pixels. Στην συνέχεια, συγκρίνουμε την τιμή x με το αν βρίσκεται αριστερά, αν εφάπτεται, ή αν είναι δεξιά του μέσου της οθόνης (τιμή μικρότερη, ίση, ή μεγαλύτερη του 160) και βάση αυτού, στρίβουμε το Maqueen plus αριστερά, το κρατάμε ευθεία ή το στρίβουμε δεξιά κατά συνέπεια.
- `huskylens.is_learned(1);huskylens.is_appear(1, HUSKYLENSResultType_t.HUSKYLENS_RESULT_ARROW),x=huskylens.read_arrow(1, Content2.X_TARGET)`. Με αυτές τις γραμμές κώδικα, λαμβάνουμε από το huskylens την οριζόντια τιμή του πέρατος της γραμμής στην οθόνη και την αποθηκεύουμε στην μεταβλητή x
- Στην συνέχεια έχουμε διαδοχικά συνθήκες, όπου συγκρίνουμε την τιμή x το μέσο της οθόνης σε pixel που είναι το 160 και οδηγούμε το όχημα αναλόγως
- Για $x = 160$ θέτουμε και τους δύο τροχούς στην ίδια ταχύτητα, οπότε το όχημα προχωράει ευθεία
- Για $x < 160$ θέτουμε και τον δεξιό τροχό σε σημαντικά μεγαλύτερη τιμή από τον αριστερό, οπότε το όχημα στρίβει αριστερά.
- Για $x > 160$ θέτουμε και τον αριστερό τροχό σε σημαντικά μεγαλύτερη τιμή από τον δεξιό, οπότε το όχημα στρίβει δεξιά



Εικόνα 51. Διάγραμμα ροής 'Line follower μέσω της κάμερας HuskyLens'.

Παρατηρήσεις

- Τμηματοποιώντας την οθόνη σε περισσότερα από δύο μέρη, μπορούμε να έχουμε περισσότερες περιπτώσεις σύγκρισης. Στη συνέχεια για κάθε περίπτωση που δημιουργούμε, μπορούμε να έχουμε διαφορετικές ταχύτητες στους τροχούς. Για παράδειγμα όσο πιο πολύ στα πλαϊνά άκρα της οθόνης βρίσκεται το πέρασ της γραμμής, αυτό σηματοδοτεί μεγαλύτερη κλίση στροφής και κατά συνέπεια μπορούμε να προσαρμόσουμε την ταχύτητα των τροχών, ώστε να παίρνουν με μεγαλύτερο έλεγχο τις στροφές

- Η κλίση της κάμερας εξαρτάται από τον χάρτη και το μέγεθος των γραμμών του χάρτη, οπότε χρειάζεται προσαρμογή ανά περίπτωση.
- Τέλος η ταχύτητα των μοτέρ τόσο στην ευθεία αλλά και στις στροφές, έχει μεγάλη σημασία στο τελικό αποτέλεσμα αφού αν το όχημα κινείται πολύ γρήγορα υπάρχει πιθανότητα σε κάποια στροφή να μην γίνει έγκαιρη ανάγνωση της γραμμής και το όχημα να βγει εκτός πορείας.

5.3 Έργο 9: Παρακολούθηση αντικειμένου

Σε αυτό το έργο χρησιμοποιούμε τον αλγόριθμο παρακολούθησης αντικειμένου της ΑΙ κάμερας HuskyLens, με σκοπό να λαμβάνουμε την θέση ενός αντικειμένου με το οποίο έχουμε εκπαιδεύσει την κάμερα. Στην συνέχεια χρησιμοποιούμε αυτή την θέση για να την προβάλλουμε στην οθόνη, σαν σηματοδότη. Στο επόμενο έργο θα βασιστούμε στην ίδια τεχνική για να φτιάξουμε ένα παιχνίδι μνήμης.

Κώδικας και τεκμηρίωση

```
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_OBJECT_TRACKING)

def on_forever():
    huskylens.request()
    if huskylens.is_appear(1,
HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        if huskylens.reade_box(1, Content1.X_CENTER) <
160:
            if huskylens.reade_box(1, Content1.Y_CENTER)
< 120:
                images.create_image("""
                    # # . . .
                    # # . . .
                    . . . . .
                    . . . . .
                    . . . . .
                    """).show_image(0)
            else:
                images.create_image("""
                    . . . . .
                    . . . . .
                    . . . . .
                    # # . . .
                    # # . . .
                    """).show_image(0)
        elif huskylens.reade_box(1, Content1.Y_CENTER) <
120:
            images.create_image("""
                . . . # #
```

```

    . . . # #
    . . . . .
    . . . . .
    . . . . .
    . . . . .
    """).show_image(0)
else:
    images.create_image("""
    . . . . .
    . . . . .
    . . . . .
    . . . . # #
    . . . . # #
    """).show_image(0)
led.set_brightness(0)
led.fade_in(100)
basic.pause(50)
led.fade_out(100)
basic.forever(on_forever)

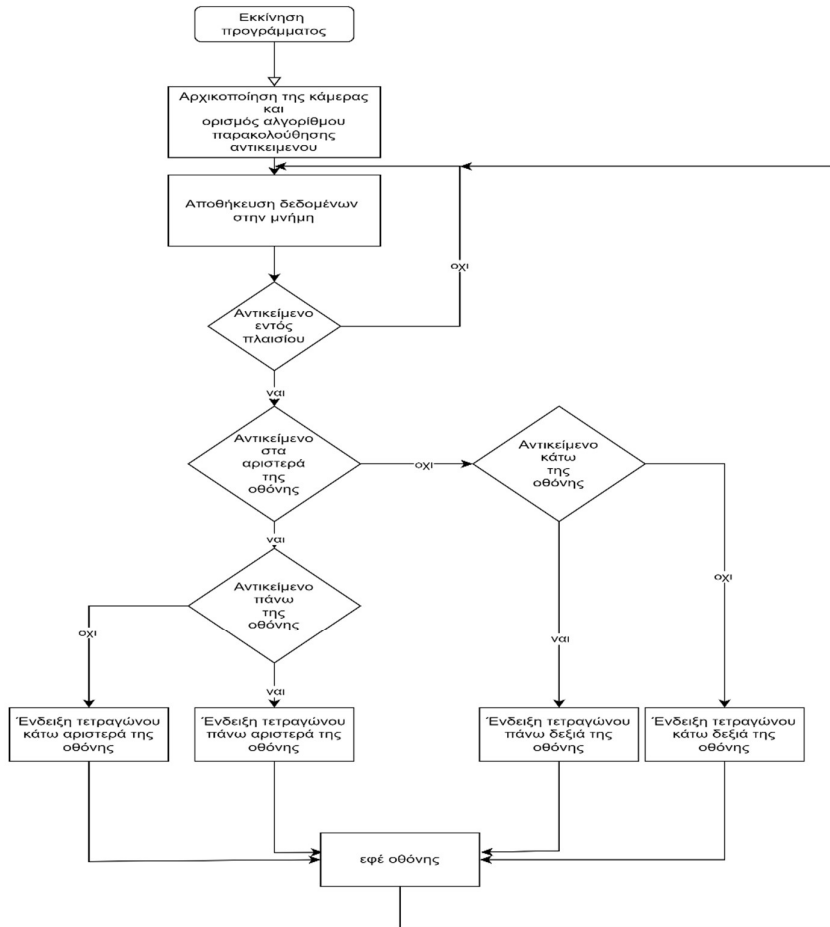
```



Εικόνα 52. Κώδικας WordBlocks 'Παρακολούθηση αντικειμένου'

Τεκμηρίωση κώδικα:

- `huskylens.init_i2c()` Αρχικοποίηση της κάμερας HuskyLens
- `huskylens.init_mode(protocolAlgorithm.ALGORITHM_OBJECT_TRACKING)` Σε αυτή τη γραμμή επιλέγουμε τον αλγόριθμο παρακολούθησης γραμμών
- Σε κάθε κύκλο του προγράμματος, η κάμερα ξεκινάει λαμβάνοντας δεδομένα από την εικόνα εκείνη την στιγμή
- Στην πρώτη συνθήκη, εξετάζουμε εάν το αντικείμενο που εκπαιδεύσαμε την κάμερα βρίσκεται εντός του πλαισίου.
- Στην δεύτερη συνθήκη εξετάζουμε αν η θέση του κέντρου του πλαισίου επάνω στο αντικείμενο είναι μικρότερη από το μέσο της οθόνης ως προς τον οριζόντιο άξονα.
- Στις εμφολευμένες συνθήκες εξετάζουμε αν η θέση του κέντρου του πλαισίου επάνω στο αντικείμενο είναι μικρότερη από το μέσο της οθόνης ως προς τον κάθετο άξονα.
- Ανάλογα με το που βρίσκεται το πλαίσιο, σχηματίζουμε ένα τετράγωνο στην εκάστοτε θέση της οθόνης του `micro:bit`
- Στις τελευταίες γραμμές 40-43 του προγράμματος δημιουργούμε ένα εφέ στα LEDs την οθόνης του `micro:bit`, αυξομειώνοντας το επίπεδο φωτός των LEDs και κάνοντας παύση του προγράμματος για ορισμένα `milliseconds`.



Εικόνα 53. Διάγραμμα ροής 'Παρακολούθηση αντικειμένου'.

Παρατηρήσεις

Για το συγκεκριμένο έργο ορίζουμε την εκμάθηση μόνο ενός αντικείμενου στις ρυθμίσεις της κάμερας. Φροντίζουμε το πλαίσιο να υπερκαλύπτει το αντικείμενο για το οποίο έχουμε εκπαιδεύσει την κάμερα (πχ ένα ζάρι).

5.4 Έργο 10: Παιχνίδι μνήμης

Στον παρακάτω έργο αξιοποιούμε τις τεχνικές του προηγούμενου έργου για να φτιάξουμε ένα παιχνίδι μνήμης. Αρχικά δείχνουμε στον παίκτη την τοποθεσία στην οποία θα πρέπει να μετακινήσει το αντικείμενο. Σε κάθε γύρο, προστίθεται και μια νέα τοποθεσία, οπότε ο παίκτης θα πρέπει διαδοχικά να τοποθετεί το αντικείμενο σε όλες τις θέσεις που του δείχνουμε. Αξιοποιούμε και πάλι τον αλγόριθμο παρακολούθησης αντικείμενου, ώστε να διαπιστώσουμε εάν ο παίκτης τοποθετεί το αντικείμενο στις σωστές θέσεις.

Κώδικας και τεκμηρίωση

```
def PlayerMove():
    global playerChoice
    huskylens.request()
    if huskylens.is_appear(1,
HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        if huskylens.reade_box(1, Content1.X_CENTER) <
160:
            if huskylens.reade_box(1, Content1.Y_CENTER)
< 120:
                images2[0].show_image(0)
                playerChoice = 0
            else:
                images2[1].show_image(0)
                playerChoice = 1
        elif huskylens.reade_box(1, Content1.Y_CENTER) <
120:
            images2[2].show_image(0)
            playerChoice = 2
        else:
            images2[3].show_image(0)
            playerChoice = 3
    currentLevel = 0
    index = 0
    playerChoice = 0
    images2: List[Image] = []
    huskylens.init_i2c()
    huskylens.init_mode(protocolAlgorithm.ALGORITHM_OBJECT_TRACKING)
    images2 = [images.create_image("""
```

```
        # # . . .
        # # . . .
        . . . . .
        . . . . .
        . . . . .
        """),
images.create_image("""
. . . . .
. . . . .
. . . . .
# # . . .
# # . . .
"""),
images.create_image("""
. . . # #
. . . # #
. . . . .
. . . . .
. . . . .
"""),
images.create_image("""
. . . . .
. . . . .
. . . # #
. . . # #
"""),
images.create_image("""
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
""")
levels = [0, 1, 3, 2, 1, 1, 0, 1, 3, 2]
playerChoice = -1

def on_forever():
    global index, currentLevel
    index = 0
    if currentLevel <= 5:
        while index <= currentLevel:
            images2[levels[index]].show_image(0)
            index += 1
            led.set_brightness(0)
            basic.pause(1000)
            led.set_brightness(255)
        images2[4].show_image(0)
        index = 0
        images.create_image("""
# # # # #
""")
```

```
        # . . . #
        # . . . #
        # . . . #
        # # # # #
        """).show_image(0)
basic.pause(2000)
while index <= currentLevel:
    for index2 in range(20):
        PlayerMove()
        basic.show_string("" + str(playerChoice))
        basic.pause(1000)
        images2[4].show_image(0)
        if playerChoice == levels[index]:
            images.create_image("""
                . . . . .
                . # . # .
                . . . . .
                . # . # .
                . . # . .
                """).show_image(0)
        else:
            images.create_image("""
                . . . . .
                . # . # .
                . . . . .
                . # # # .
                . # . # .
                """).show_image(0)
            basic.pause(1000)
            break
        basic.pause(1000)
        index += 1
        currentLevel += 1
        music.play(music.tone_playable(262,
music.beat(BeatFraction.WHOLE)),
        music.PlaybackMode.UNTIL_DONE)
        basic.pause(2000)
    else:
        images.create_image("""
        # # # # #
        # # # # #
        # # # # #
        # # # # #
        # # # # #
        """).show_image(0)
basic.forever(on_forever)
```

```

on start
  Huskylens initialize I2C until success
  Huskylens switch algorithm to Object Tracking

  array of
    create image
    create image
    create image
  set images2 to
    create image
    create image
    create image
  set levels to
    array of
      0
      1
      1
      2
      1
      1
      1
      2
  set playerChoice to 1

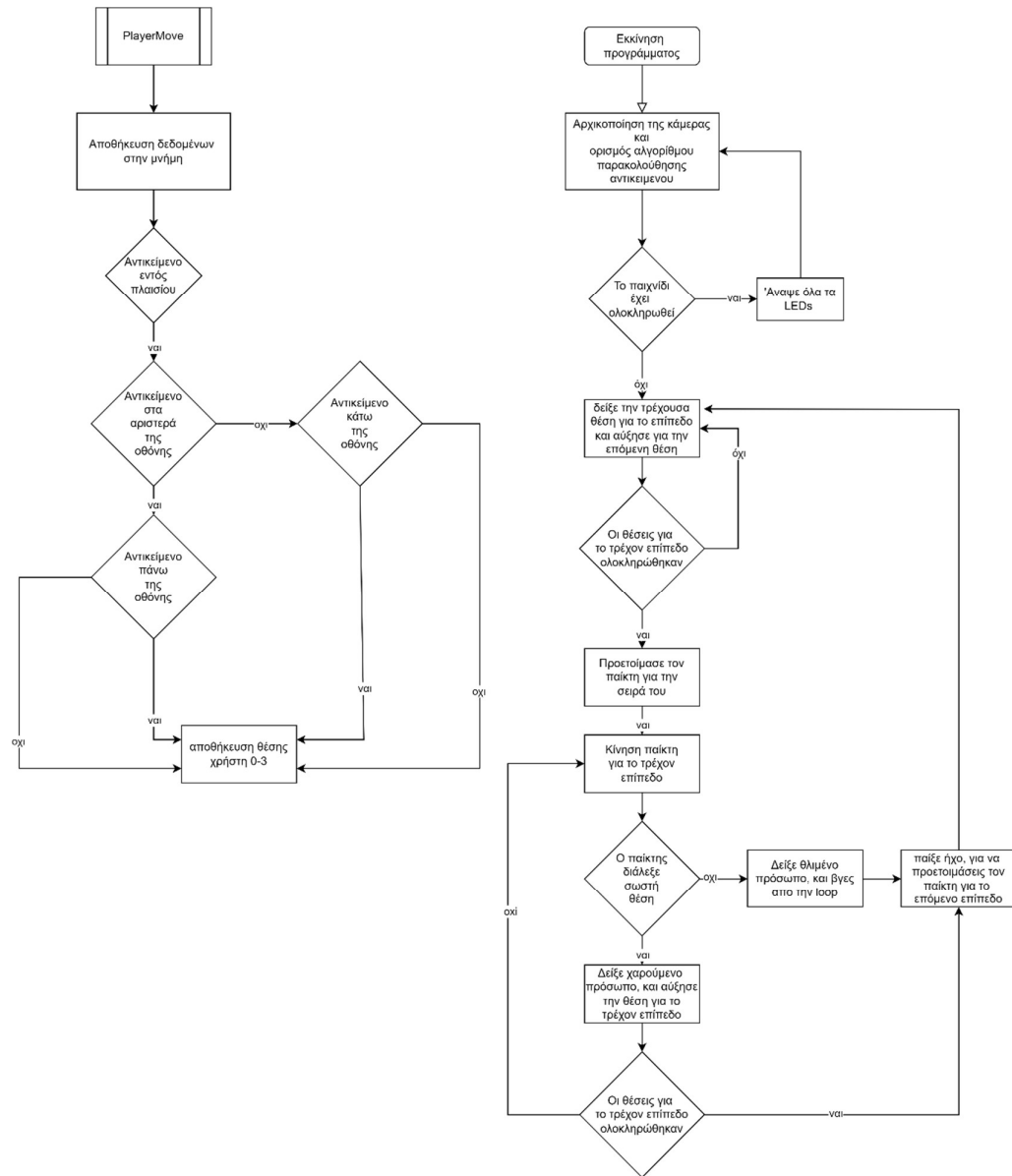
function PlayerMove
  Huskylens request data once and save into the result
  if Huskylens check if ID 1 frame is on screen from the result then
    if Huskylens get X center of ID 1 frame from the result < 160 then
      if Huskylens get Y center of ID 1 frame from the result < 120 then
        show image images2 get value at 0 at offset 0
        set playerChoice to 0
      else
        show image images2 get value at 1 at offset 0
        set playerChoice to 1
    else if Huskylens get Y center of ID 1 frame from the result < 120 then
      show image images2 get value at 2 at offset 0
      set playerChoice to 2
    else
      show image images2 get value at 3 at offset 0
      set playerChoice to 3

forever
  set index to 0
  if currentLevel <= 5 then
    while index < currentLevel
      do
        show image images2 get value at levels get value at index at offset 0
        change index by 1
        set brightness 0
        pause (ms) 1000
        set brightness 255
      show image images2 get value at 4 at offset 0
      set index to 0
      create image
      show image at offset 0
      pause (ms) 2000
      while index < currentLevel
        do
          repeat 20 times
            do
              call PlayerMove
            show string join playerChoice
            pause (ms) 1000
            show image images2 get value at 4 at offset 0
          if playerChoice = levels get value at index then
            create image
            show image at offset 0
          else
            create image
            show image at offset 0
            pause (ms) 1000
            break
          pause (ms) 1000
          change index by 1
        change currentLevel by 1
        play tone Middle C for 1 beat until done
        pause (ms) 2000
      else
        create image
        show image at offset 0
  
```

Εικόνα 54. Κώδικας WordBlocks 'Παιχνίδι μνήμης'

Τεκμηρίωση κώδικα:

- Στην συνάρτηση onStart αρχικοποιούμε την κάμερα, ορίζουμε τον αλγόριθμο παρακολούθησης γραμμών και επίσης ορίζουμε δύο πίνακες. Ο πρώτος έχει αντικείμενα εικόνας για την οθόνη του micro:bit. Η κάθε θέση του δεύτερου πίνακα αντιστοιχεί στο κάθε επίπεδο του παιχνιδιού (σύνολο δέκα επίπεδα) και περιλαμβάνει από 0 έως 3 που είναι η αντιστοίχιση της θέσης στα τέσσερα τμήματα της οθόνης. (0 = πάνω αριστερά, 1=κάτω αριστερά, 2=πάνω δεξιά, 3=κάτω δεξιά)
- Η συνάρτηση PlayerMove χρησιμοποιεί την ίδια λογική με το προηγούμενο έργο, με την διαφορά ότι αντί να δείχνουμε κάποια εικόνα, αποθηκεύουμε την θέση του αντικειμένου ως προς τα τέσσερα τμήματα της οθόνης 0-3. Συγκεκριμένα, σε κάθε κλήση της συνάρτησης, η κάμερα ξεκινάει λαμβάνοντας δεδομένα από την εικόνα εκείνη την στιγμή. Στην πρώτη συνθήκη, εξετάζουμε εάν το αντικείμενο που εκπαιδεύσαμε την κάμερα βρίσκεται εντός του πλαισίου. Στις επόμενες συνθήκες αποθηκεύουμε την θέση του αντικειμένου, ανάλογα με το αν βρίσκεται σε ένα από τα τέσσερα τμήματα της οθόνης.
- Μέσα στην συνάρτηση forever, γίνεται όλη η λογική του παιχνιδιού. Σε κάθε κύκλο εξετάζουμε το επίπεδο που βρίσκεται ο χρήστης. Αν θέλουμε περισσότερα επίπεδα πρέπει να τα ορίσουμε τόσο στον πίνακα levels καθώς και να ορίσουμε την τιμή σύγκρισης της μεταβλητής currentLevel.
- Στον while βρόγχο της γραμμής 66, δείχνουμε στον παίκτη όλες τις τοποθεσίες του επιπέδου που βρισκόμαστε αυτήν την στιγμή. Ταυτόχρονα, χρησιμοποιούμε μερικές παύσεις καθώς και καθαρίζουμε την οθόνη, ώστε ο χρήστης να έχει μια καλή εμπειρία χρήσης.
- Στις γραμμές 72-80 δείχνουμε στον παίκτη ένα τετράγωνο και περιμένουμε δύο δευτερόλεπτα, ώστε να τον προετοιμάσουμε ότι είναι η σειρά του και πρέπει να κινηθεί
- Στον while βρόγχο της γραμμής 82, εξετάζουμε τις κινήσεις του παίκτη για το κάθε επίπεδο. Αρχικά λαμβάνουμε την θέση του αντικειμένου που φαίνεται στην οθόνη. Καλούμε την συνάρτηση πολλές φορές, ούτως ώστε να διαβεβαιώσουμε ότι υπάρχει αρκετός χρόνος ώστε η κάμερα να λάβει τα σωστά δεδομένα. Μόλις γίνει αυτό, τυπώνουμε την θέση που διάλεξε ο χρήστης, για να έχει επιβεβαίωση της κίνησης του.
- Στην συνέχεια, γίνεται έλεγχος για να διαπιστωθεί εάν η επιλογή του παίκτη συμφωνεί με την εκάστοτε θέση διαδοχικά, του κάθε επιπέδου. Αν ο παίκτης έχει σωστή θέση, για κάθε θέση του επιπέδου, τότε στην οθόνη θα εμφανιστεί ένα χαρωπό πρόσωπο, Στην συνέχεια το πρόγραμμα θα ηχήσει, για να προετοιμάσει τον παίκτη για το επόμενο επίπεδο. Αν κάνει λάθος, τότε θα εμφανιστεί ένα θλιμμένο πρόσωπο και αφού ηχήσει ο ήχος, το παιχνίδι θα προχωρήσει απευθείας στο επόμενο επίπεδο.
- Όταν τελειώσουν όλα τα επίπεδα, όλα τα LEDs θα ανάψουν για να μας δείξουν το τέλος του παιχνιδιού.



Εικόνα 55. Διάγραμμα ροής 'Παιχνίδι μνήμης'.

Παρατηρήσεις

Για το συγκεκριμένο έργο ορίζουμε την εκμάθηση μόνο ενός αντικειμένου στις ρυθμίσεις της κάμερας. Φροντίζουμε το πλαίσιο να υπερκαλύπτει το αντικείμενο για το οποίο έχουμε εκπαιδεύσει την κάμερα (πχ ένα ζάρι). Προσθέτοντας τιμές 0-3 στον πίνακα levels αυξάνουμε την δυσκολία και την διάρκεια του παιχνιδιού. Το πρόγραμμα, δίνει αρκετό χρόνο στον παίκτη να επιλέξει την κάθε θέση, όμως σε κάθε γύρο, θα πρέπει να περιμένει έως ότου το πρόγραμμα του δείξει τον αριθμό της θέσης, πρώτου συνεχίσει στην επόμενη του επιλογή.

5.5 Έργο 11: Μάντεψε το χρώμα

Στο παρακάτω πρόγραμμα, φτιάχνουμε ένα παιχνίδι που παίζεται με δύο παίκτες. Αρχικά ο πρώτος παίκτης επιλέγει, ένα χρώμα αριθμημένο από το ένα έως το τρία. Στην συνέχεια ο δεύτερος παίκτης έχει μία ευκαιρία να μαντέψει το σωστό χρώμα, κρύβοντας στην κάμερα την επιλογή του. Χρησιμοποιούμε, τον αλγόριθμο αναγνώρισης χρωμάτων της κάμερας, καθώς επίσης και τα πλήκτρα του micro:bit αρχικά για να ορίσουμε την επιλογή του πρώτου παίκτη και επίσης για να ξεκινήσουμε το παιχνίδι από την αρχή.

Κώδικας και τεκμηρίωση

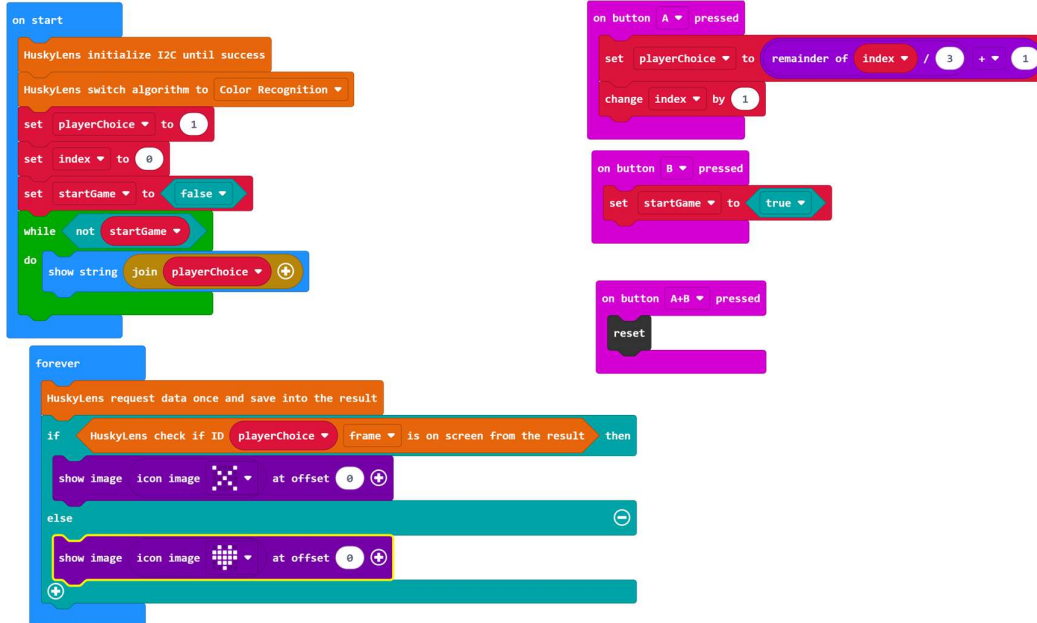
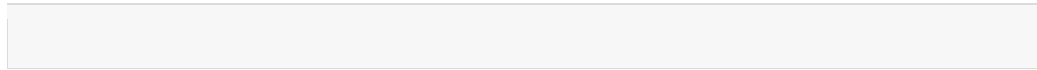
```
def on_button_pressed_a():
    global playerChoice, index
    playerChoice = index % 3 + 1
    index += 1
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_button_pressed_ab():
    control.reset()
input.on_button_pressed(Button.AB,
on_button_pressed_ab)

def on_button_pressed_b():
    global startGame
    startGame = True
input.on_button_pressed(Button.B, on_button_pressed_b)

startGame = False
index = 0
playerChoice = 0
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_COLOR_RECOGNITION)
playerChoice = 1
index = 0
startGame = False
while not (startGame):
    basic.show_string("" + str(playerChoice))

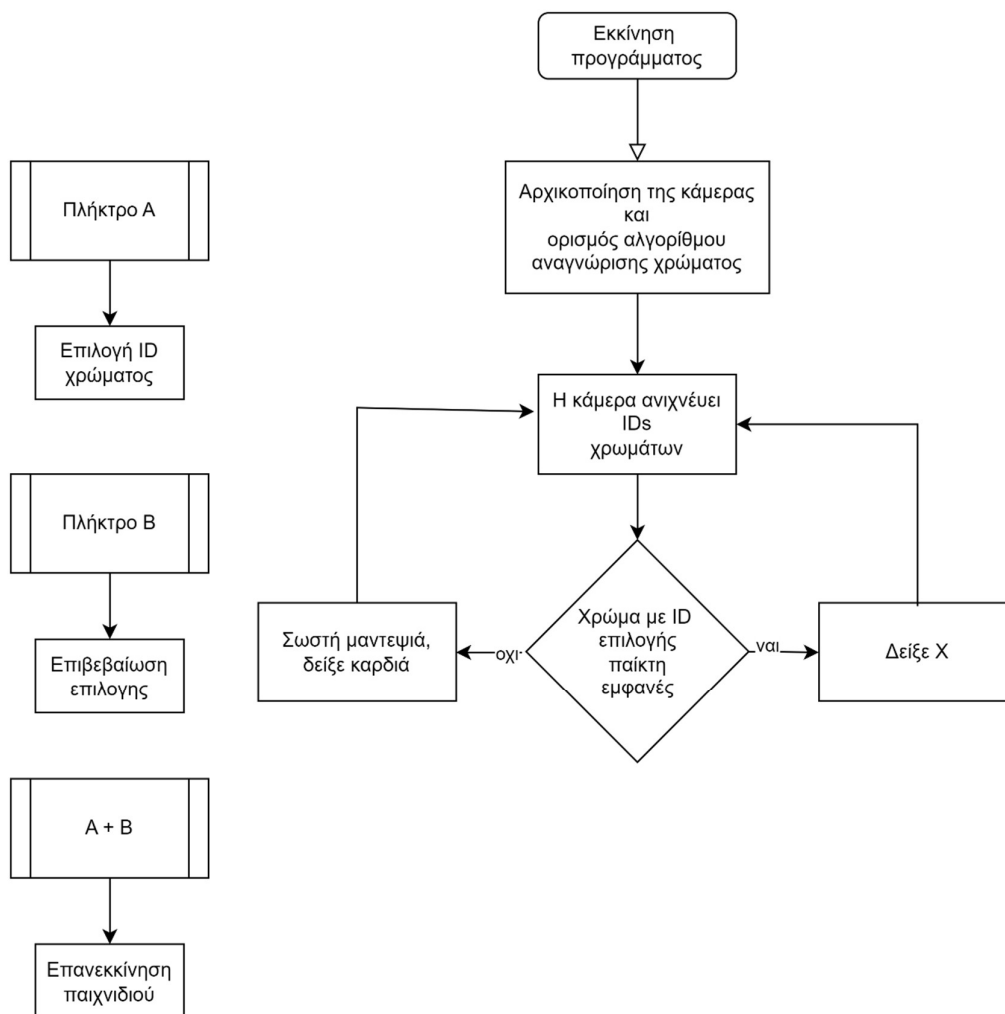
def on_forever():
    huskylens.request()
    if huskylens.is_appear(playerChoice,
HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        images.icon_image(IconNames.NO).show_image(0)
    else:
        images.icon_image(IconNames.HEART).show_image(0)
)
basic.forever(on_forever)
```

Εικόνα 56. Κώδικας WordBlocks 'Μάντεψε το χρώμα'

Τεκμηρίωση κώδικα:

- `on_button_pressed_a` Αν πατηθεί το πλήκτρο A του micro:bit, αλλάζει η επιλογή του πρώτου παίκτη. Έχοντας τρεις επιλογές στο σύνολο, χρησιμοποιούμε τον τελεστή % ώστε να κρατήσουμε το υπόλοιπο της διαίρεσης μιας αριθμητικής μεταβλητής (που ξεκινάει από το 0 και αυξάνεται κατά ένα σε κάθε πάτημα του πλήκτρου A) με το τρία. Το υπόλοιπο της διαίρεσης το αυξάνουμε κατά 1 και το αποθηκεύουμε στην επιλογή του πρώτου παίκτη που αντιστοιχεί στα χρώματα 1-3.
- `on_button_pressed_b` Μετά την επιλογή του χρώματος ο πρώτος παίκτης πατάει το B, ώστε να επιβεβαιώσει την επιλογή του και να συνεχίσει ο δεύτερος παίκτης.
- `on_button_pressed_ab` Αν πατηθούν ταυτόχρονα και τα δυο πλήκτρα, το παιχνίδι θα αρχίσει από την αρχή.
- `huskylens.init_i2c()``huskylens.init_mode(protocolAlgorithm.ALGORITHM_COLOR_RECOGNITION)` Αρχικοποίηση της κάμερας, ορισμός του αλγορίθμου αναγνώρισης χρώματος και ορισμός μεταβλητών
- στην συνάρτηση `on_forever` γίνεται έλεγχος της κάμερας για δεδομένα
- Έως ότου ο χρήστης να κρύψει το σωστό χρώμα θα εμφανίζεται στην οθόνη το X. Στην περίπτωση που επιλέξει και κρύψει το σωστό, θα εμφανιστεί καρδιά



Εικόνα 57. Διάγραμμα ροής 'Μάντεψε το χρώμα'.

Παρατηρήσεις

Για το συγκεκριμένο έργο ορίζουμε την εκμάθηση πολλαπλών χρωμάτων. Στην περίπτωση που θέλουμε να κάνουμε δυσκολότερο το παιχνίδι, ορίζουμε περισσότερα χρώματα, και αυξάνουμε τον αριθμό των χρωμάτων στον κώδικα.

5.6 Έργο 12: Εφέ μουσικής

Στο πρόγραμμα εφέ μουσικής χρησιμοποιούμε και πάλι τον αλγόριθμο αναγνώρισης χρωμάτων. Βάση χρώματος που η κάμερα βλέπει στον χώρο, θα αλλάζει τα χρώματα των RGB LEDs του Maqueen Plus. Επίσης, το μικρόφωνο του micro:bit θα αντιδρά στον ήχο, και θα αναβοσβήνει ρυθμικά τα φώτα της οθόνης καθώς και τα εμπρόσθια φώτα του Maqueen Plus. Συνεπώς πρόκειται για ένα φωτορυθμικό, όπως βρίσκουμε σε διάφορα gadgets του εμπορίου.

Κώδικας και τεκμηρίωση

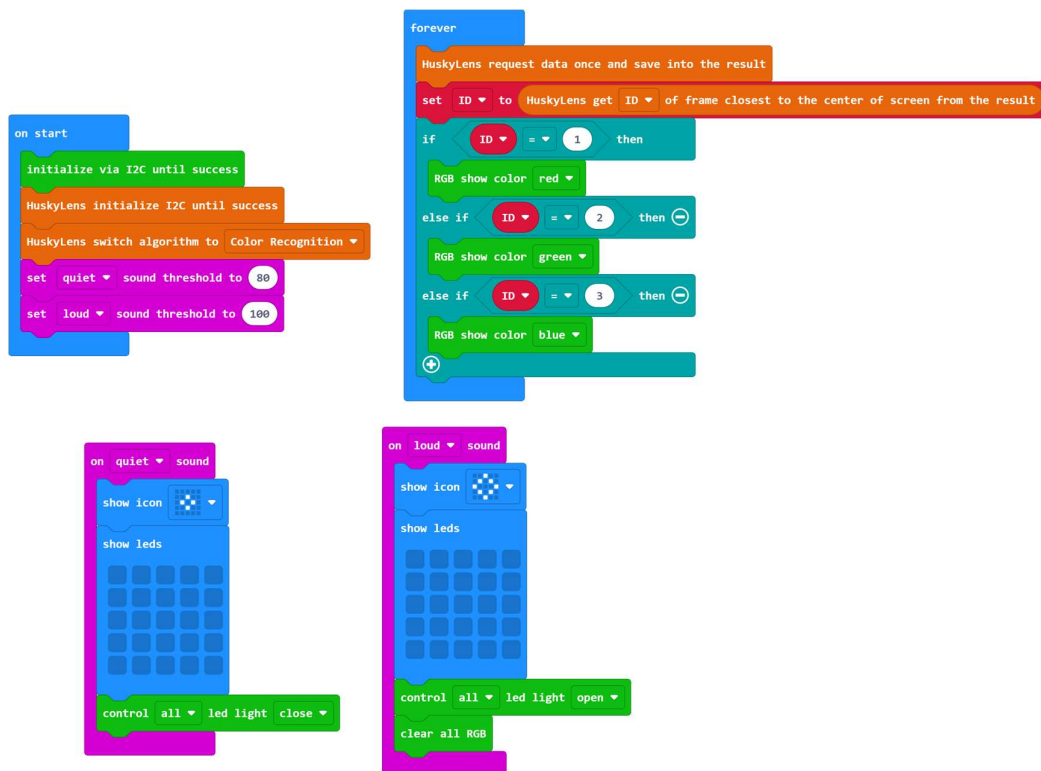
```
def on_sound_loud():
    basic.show_icon(IconNames.DIAMOND)
    basic.show_leds("""
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        """)
    maqueenPlusV2.control_led(maqueenPlusV2.MyEnumLed.ALL_LED,
                              maqueenPlusV2.MyEnumSwitch.OPEN)
    maqueenPlusV2.led_blank()
input.on_sound(DetectedSound.LOUD, on_sound_loud)

def on_sound_quiet():
    basic.show_icon(IconNames.SMALL_DIAMOND)
    basic.show_leds("""
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        """)
    maqueenPlusV2.control_led(maqueenPlusV2.MyEnumLed.ALL_LED,
                              maqueenPlusV2.MyEnumSwitch.CLOSE)
input.on_sound(DetectedSound.QUIENT, on_sound_quiet)

ID = 0
maqueenPlusV2.i2c_init()
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_COLOR_RECOGNITION)
input.set_sound_threshold(SoundThreshold.QUIENT, 80)
input.set_sound_threshold(SoundThreshold.LOUD, 100)

def on_forever():
    global ID
```

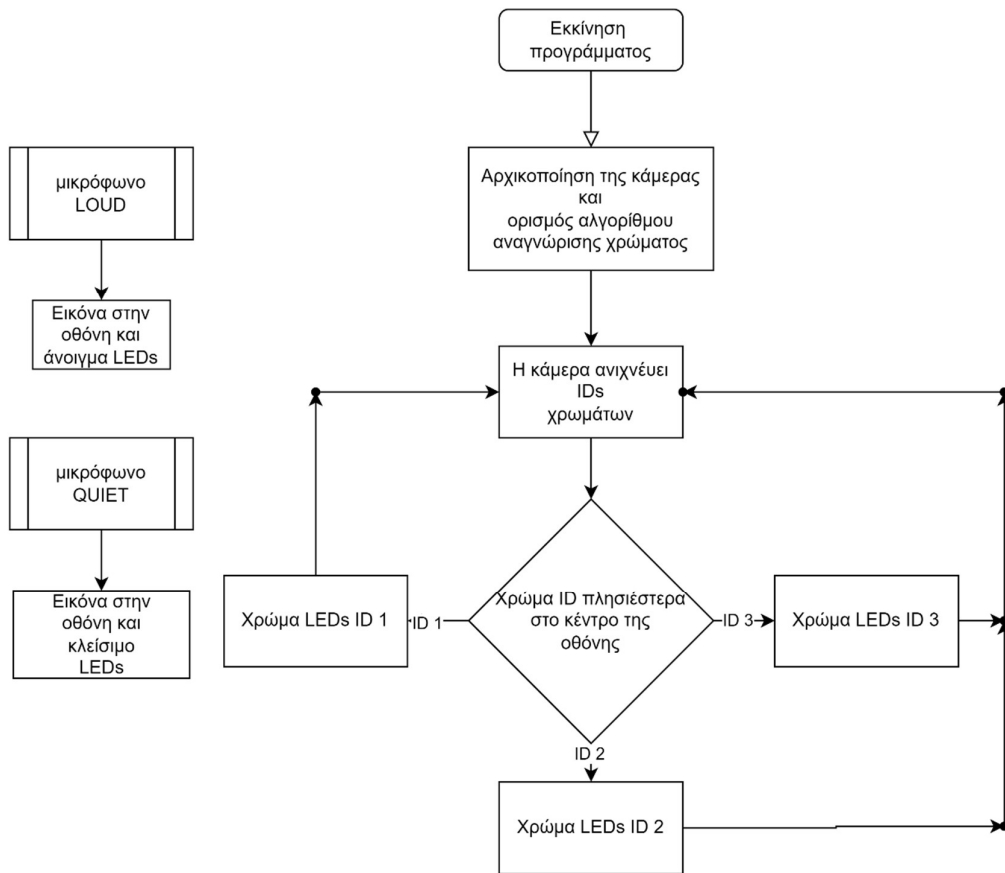
```
huskylens.request()
ID = huskylens.readBox_s(Content3.ID)
if ID == 1:
    maqueenPlusV2.show_color(maqueenPlusV2.NeoPixelColors.RED)
elif ID == 2:
    maqueenPlusV2.show_color(maqueenPlusV2.NeoPixelColors.GREEN)
elif ID == 3:
    maqueenPlusV2.show_color(maqueenPlusV2.NeoPixelColors.BLUE)
basic.forever(on_forever)
```



Εικόνα 58. Κώδικας WordBlocks 'Εφέ μουσικής'

Τεκμηρίωση κώδικα:

- Στις συναρτήσεις `on_sound_loud()` και `on_sound_quiet()` εμγανίζουμε σχήματα στην οθόνη και μέσω του `maqueenPlusV2.control_led` ανάβουμε ή σβήνουμε τα εμπρόσθια φώτα του Maqueen Plus
- Στην συνέχεια κάνουμε αρχικοποίηση παραμέτρων καθώς και ορίζουμε τον αλγόριθμο αναγνώρισης χρώματος για την κάμερα καθώς επίσης ορίζουμε το κατώφλι έντασης μικροφώνου για τις περιπτώσεις QUIT και LOUD (0-255)
- Στην συνάρτηση `on_forever()` Η κάμερα λαμβάνει δεδομένα και ανάλογα με το αν το πλαίσιο που βρίσκεται πιο κοντά στο κέντρο της οθόνης έχει κάποια τιμή ID ίδιο με κάποιο από αυτά που έχουμε ορίσει, τα LEDs του Maqueen Plus θα ανάψουν με το ανάλογο χρώμα



Εικόνα 59. Διάγραμμα ροής 'Εφέ μουσικής'.

Παρατηρήσεις

Για το συγκεκριμένο έργο ορίζουμε την εκμάθηση πολλαπλών χρωμάτων. Όσα περισσότερα χρώματα εκπαιδύσουμε την κάμερα και ορίσουμε στο πρόγραμμα, το gadget θα ανταποκρίνεται καλύτερα στο περιβάλλον και τον χώρο.

5.7 Έργο 13: Πράσινο κόκκινο φως

Το επόμενο πρόγραμμα βασίζεται στο κλασικό ομώνυμο παιχνίδι, στο οποίο ένας παίκτης γυρισμένος προς τους υπόλοιπους μετράει αντίστροφα και όταν γυρίσει οι υπόλοιποι παίκτες πρέπει να μην κινούνται. Τον ρόλο αυτό, τον αναλαμβάνει το πρόγραμμα, το οποίο με την κάμερα διαπιστώνει αν ο παίκτης κινείται. Σκοπός του παίκτη είναι να καταφέρει να πατήσει το πλήκτρο A χωρίς να τον εντοπίσει η κάμερα. Σε αυτό το έργο γίνεται χρήση του αλγορίθμου αναγνώρισης προσώπων.

Κώδικας και τεκμηρίωση

```
def checkIfFaceInBounds():
    global isInbounds
    huskylens.request()
    if huskylens.is_appear(1,
HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        if huskylens.reade_box(1, Content1.X_CENTER) >
140 and huskylens.reade_box(1, Content1.X_CENTER) < 180:
            isInbounds = True
        else:
            isInbounds = False

def on_button_pressed_a():
    global isWin, gameOver
    isWin = True
    gameOver = True
input.on_button_pressed(Button.A, on_button_pressed_a)

def blink():
    images.icon_image(IconNames.SQUARE).show_image(0)
    led.set_brightness(0)
    led.fade_in(100)
    basic.pause(100)
    led.fade_out(100)

def on_button_pressed_b():
    control.reset()
input.on_button_pressed(Button.B, on_button_pressed_b)

gameOver = False
isWin = False
isInbounds = False
huskylens.init_i2c()
huskylens.init_mode(protocolAlgorithm.ALGORITHM_FACE_RECO
GNITION)
isInbounds = True
isWin = False
gameOver = False

def on_forever():
    global gameOver
    while not (gameOver):
        for index in range(3):
            music.play(music.tone_playable(262,
music.beat(BeatFraction.HALF)),
                    music.PlaybackMode.UNTIL_DONE)
            blink()
            led.set_brightness(255)
            music.play(music.tone_playable(523,
music.beat(BeatFraction.WHOLE)),
```

```
        music.PlaybackMode.UNTIL_DONE)
    basic.pause(1000)
    if not (gameOver):
        for index2 in range(10):
            checkIfFaceInBounds()
            if isInbounds:
                images.icon_image(IconNames.HAPPY).sh
ow_image(0)
            else:
                gameOver = True
    if gameOver and isWin:
        images.icon_image(IconNames.YES).show_image(0)
    else:
        images.icon_image(IconNames.NO).show_image(0)
basic.forever(on_forever)
```

```

on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Face Recognition
  set isInbounds to true
  set isMin to false
  set gameOver to false

function checkIfFaceInBounds
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    if HuskyLens get X center of ID 1 frame from the result > 140 and HuskyLens get X center of ID 1 frame from the result < 180 then
      set isInbounds to true
    else
      set isInbounds to false

function blink
  show image icon image at offset 0
  set brightness 0
  <python code>
  pause (ms) 100
  <python code>

while not gameOver
  do
    repeat 3 times
      do
        play tone Middle C for 1/2 beat until done
        call blink
        set brightness 255
      play tone High C for 1 beat until done
      pause (ms) 1000
      if not gameOver then
        repeat 10 times
          do
            call checkIfFaceInBounds
            if isInbounds then
              show image icon image at offset 0
            else
              set gameOver to true
          if gameOver and isMin then
            show image icon image at offset 0
          else
            show image icon image at offset 0

on button A pressed
  set isMin to true
  set gameOver to true

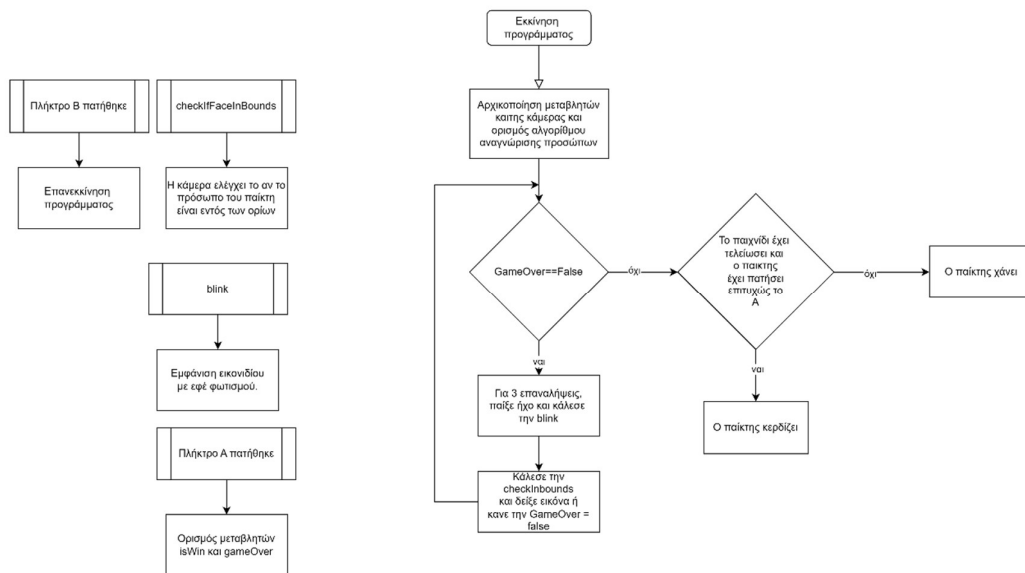
on button B pressed
  reset
  
```

Εικόνα 60. Κώδικας WordBlocks 'Πράσινο κόκκινο φως'

Τεκμηρίωση κώδικα:

- Στην συνάρτηση checkIfFaceInBounds η κάμερα λαμβάνει δεδομένα και στην συνέχεια, εάν το κέντρο του πλαισίου που καλύπτει το πρόσωπο βρίσκεται εντός των ορίων που έχουμε ορίσει ως προς το οριζόντιο άξονα, η μεταβλητή isInbounds θα γίνει αληθής ή ψευδής αντίστοιχα.
- Στην συνάρτηση on_button_pressed_a οι μεταβλητές isWin και gameOver γίνονται αληθής, για να υποδηλώσουν ότι ο παίκτης έχει κερδίσει.
- Στην συνάρτηση blink γίνεται εμφάνιση μιας εικόνας στα LEDs του micro:bit με κάποιο εφέ που αναβοσβήνει.
- Αν πατηθεί το πλήκτρο B γίνεται επανεκκίνηση προγράμματος
- Στις γραμμές 27 με 34 γίνεται αρχικοποίηση μεταβλητών του προγράμματος και ορισμός του αλγορίθμου της κάμερας σε αναγνώριση προσώπων.

- Στην συνάρτηση on_forever γίνεται η βασική λογική του παιχνιδιού. Αρχικά όσο το παιχνίδι δεν έχει ολοκληρωθεί (gameOver=false_ συμβαίνουν τα ακόλουθα. Παίζει ένας ήχος 3 φορές μέσω της συνάρτησης music.play και επίσης αναβοσβήνει ένα τετράγωνο σχήμα στην οθόνη. Αυτό γίνεται για να καταλάβει ο παίκτης πως κατά την διάρκεια αυτή, μπορεί να κινηθεί προς την κάμερα. Ο τελευταίος ήχος σηματοδοτεί στον παίκτη πως πρέπει να ακινητοποιηθεί.
- Στην επανάληψη for της γραμμής 47-53 καλούμε την checkIfFaceInBounds ώστε να διαπιστώσουμε αν ο παίκτης είναι κινητοποιημένος. Αν κατά την διάρκεια αυτή ο παίκτης κινηθεί σημαντικά, τότε η μεταβλητή isInbounds θα γίνει false, με αποτέλεσμα η μεταβλητή gameOver να γίνει true. Από την άλλη, όσο η isInbounds είναι true, θα εμφανιστεί ένα χαρούμενο πρόσωπο στην οθόνη του micro:bit που υποδηλώνει ότι το παιχνίδι συνεχίζει.
- Όταν η gameOver γίνει false, ο βρόγχος while θα σταματήσει. Στην συνέχεια θα διαπιστωθεί αν ο παίκτης κέρδισε ή όχι δεδομένου ότι έχει πατήσει το πλήκτρο A και η τιμή isWin έγινε true και θα εμφανιστεί κατάλληλο εικονίδιο στην οθόνη του micro:bit



Εικόνα 61. Διάγραμμα ροής 'Πράσινο κόκκινο φως'.

Παρατηρήσεις

Σημαντικό είναι κατά την εκκίνηση του παιχνιδιού, ο παίκτης να βρίσκεται ελάχιστα μέτρα μακριά από την κάμερα, όχι όμως υπερβολικά πολύ, αφού ενδέχεται η κάμερα να μην μπορεί να τον εντοπίσει. Επίσης ξεκινώντας ο παίκτης πρέπει να βρίσκεται κατά το δυνατόν στην μέση της κάμερας.

6. Συμπεράσματα

Η εργασία σχετικά με την εκπαιδευτική ρομποτική, χρησιμοποιώντας το Microbit, την ΑΙ κάμερα Huskylens και το Maqueen Plus, αναδεικνύει τη σημασία της τεχνολογίας στην εκπαίδευση. Μέσω αυτών των εκπαιδευτικών εργαλείων, οι μαθητές αποκτούν εμπειρία στον τομέα της ρομποτικής και του προγραμματισμού, αναπτύσσοντας παράλληλα τη δημιουργικότητά τους και τον κριτικό στοχασμό τους.

Παρά τη χρήση αυτών των εκπαιδευτικών κιτ, είναι σημαντικό να σημειωθεί ότι για εφαρμογές που απαιτούν μεγαλύτερη ακρίβεια, τα εν λόγω εργαλεία μόνα τους δεν επαρκούν. Ωστόσο, για τον σκοπό της εκπαίδευσης και της εξοικείωσης των μαθητών με την τεχνολογία, αποτελούν απαραίτητο εργαλείο. Μέσω της εκπαιδευτικής ρομποτικής, οι μαθητές μαθαίνουν να λύνουν προβλήματα, να συνεργάζονται και να εξερευνούν τις δυνατότητες της τεχνολογίας με διασκεδαστικό και δημιουργικό τρόπο.

Για την σωστή εκτέλεση των έργων, φροντίζουμε τα παρακάτω:

1. Σημασία της απόστασης και της θέσης από την κάμερα
2. Επαρκής φωτισμός: Η εκπαίδευση της κάμερας πρέπει να γίνεται σε ένα αρκετά φωτεινό περιβάλλον.
3. Ανίχνευση ενός αντικειμένου: Η εστίαση σε ένα μόνο αντικείμενο μπορεί να απλοποιήσει τον χειρισμό του συστήματος, ώστε η κάμερα να αποδίδει καλύτερα.
4. Προσαρμογή της γραμμής και της ταχύτητας: Η κατάλληλη προσαρμογή της γραμμής(διαστάσεις) και της ταχύτητας των τροχών του Maqueen Plus είναι απαραίτητη για τη σωστή λειτουργία και την αποφυγή σφαλμάτων.

Βιβλιογραφία και ιστότοποι

- [1] S. Russell and P. Norvig Artificial Intelligence: A Modern Approach (2nd edition),
Prentice Hall, 2nd ed., 2003
- [2] <https://edu.dfrobot.com/> Τελευταία Πρόσβαση: 28/02/2024
- [3] <https://makecode.microbit.org/> Τελευταία Πρόσβαση: 28/02/2024
- [4] https://en.wikipedia.org/wiki/History_of_robots Τελευταία Πρόσβαση:
28/02/2024
- [5] https://en.wikipedia.org/wiki/Machine_learning Τελευταία Πρόσβαση:
28/02/2024
- [6] <https://www.unesco.org/en/artificial-intelligence/recommendation-ethics>
Τελευταία Πρόσβαση: 28/02/2024
- [7] <https://microbit.org/> Τελευταία Πρόσβαση: 28/02/2024
- [8] https://en.wikipedia.org/wiki/Micro_Bit Τελευταία Πρόσβαση: 28/02/2024
- [9] <https://www.arduino.cc/en/Guide/Introduction> Τελευταία Πρόσβαση: 28/02/2024
- [10] <https://www.python.org/> Τελευταία Πρόσβαση: 28/02/2024
- [11] [https://developer.mozilla.org/en-
US/docs/Learn/JavaScript/First_steps/What_is_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript) Τελευταία Πρόσβαση:
28/02/2024
- [12] <https://education.lego.com/en-us/> Τελευταία Πρόσβαση: 28/02/2024
- [13] <https://files.eric.ed.gov/fulltext/ED537186.pdf> Τελευταία Πρόσβαση: 28/02/2024