



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
Τμ. ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ  
Π.Μ.Σ. «ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΚΑΙ  
ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΑ ΣΥΣΤΗΜΑΤΑ»

---

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΚΑΙ ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΑ  
ΣΥΣΤΗΜΑΤΑ»

ΤΙΤΛΟΣ

*Εξέλιξη και βελτίωση ρομποτικού συστήματος για την πρωτοβάθμια  
εκπαίδευση*

---

ΤΙΤΛΟΣ ΑΓΓΛΙΚΑ

*Development and improvement of robotic system for primary education*

---

Όνοματεπώνυμο Φοιτητή:

*Ιορδανίδης Ιορδάνης*

---

Όνοματεπώνυμο Υπεύθυνου Καθηγητή:

*Χατζόπουλος Αβραάμ*

---

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΑΡΤΙΟΣ 2024

---

Μέλη Εξεταστικής Επιτροπής

---

---

---

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ιορδανίδης Ιορδάνης του Νικολάου, με αριθμό μητρώου 8096631 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «Μη Επανδρωμένα Αυτόνομα και Τηλεκατευθυνόμενα Συστήματα» του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής της Σχολής Μηχανικών Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι: «Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου».

Ο δηλών

**Ιορδανίδης Ιορδάνης**



Ημερομηνία

**04/04/2024**

**Εξέλιξη και βελτίωση ρομποτικού συστήματος για την πρωτοβάθμια  
εκπαίδευση**

**Ιορδανίδης Ιορδάνης**

**Μεταπτυχιακή Διπλωματική Εργασία που υποβάλλεται στο καθηγητικό σώμα  
για την μερική εκπλήρωση των υποχρεώσεων απόκτησης του μεταπτυχιακού  
τίτλου του Προγράμματος Μεταπτυχιακών Σπουδών «Μη Επανδρωμένα  
Αυτόνομα και Τηλεκατευθυνόμενα Συστήματα» του Τμήματος Μηχανικών  
Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής.**

## ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία επιδιώκει τη βελτίωση ενός υφιστάμενου εκπαιδευτικού ρομποτικού οχήματος με την αναβάθμιση των δυνατοτήτων του με γνώμονα την καταλληλότητα χρήσης από μικρά παιδιά και την επίτευξη προσιτού κόστους κατασκευής.

Αρχικά, το υφιστάμενο ρομποτικό σύστημα αναλύεται με την χρήση αντίστροφης μηχανικής για την κατανόηση του, σε τέτοιο βαθμό ώστε, να μπορούν να γίνουν οι αλλαγές που επιδιώκονται μέσω της παρούσας διπλωματικής. Στη συνέχεια, αναλύονται οι διάφορες πιθανές προσθήκες που θα μπορούσαν να γίνουν, και μέσω δοκιμών εφαρμόζονται μερικές από αυτές.

Τέλος, πραγματοποιούνται δοκιμές μεταξύ των προσθηκών που έγιναν, για να επιβεβαιωθεί η σωστή λειτουργία του συστήματος. Η εν λόγω διπλωματική αποσκοπεί στην απλοποίηση της διαδικασίας εκπαίδευσης και αλληλεπίδρασης των παιδιών με το ρομπότ, ενισχύοντας την εκπαιδευτική εμπειρία τους και διευκολύνοντας την εξέλιξη των γνώσεων των παιδιών σε πολλούς τομείς, όπως την τεχνολογία, την ανάπτυξη δεξιοτήτων προγραμματισμού και την κατανόηση της ρομποτικής.

Λέξεις Κλειδιά: Πρωτοβάθμια εκπαίδευση, Εκπαιδευτική ρομποτική, Μικροελεγκτής, STEM

## **ABSTRACT**

This dissertation seeks to improve an existing educational robotic vehicle by upgrading its capabilities considering the suitability for use by young children and achieving an affordable manufacturing cost.

Initially, the existing robotic system is selected and analyzed using reverse engineering to understand it, to such an extent that the changes sought through this dissertation can be made. Then, the various possible additions that could be made are analyzed, before some of them are developed and implemented.

Finally, tests are carried out between the additions made, to confirm its correct operation of the system. The dissertation aims to simplify the process of children's education and interaction with the robot, enhancing their educational experience and facilitating the development of children's knowledge in many areas, such as technology, development of programming skills and understanding of robotics.

Keywords: Primary education, Educational robotics, Microcontroller, STEM

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	1
ABSTRACT.....	2
ΠΕΡΙΕΧΟΜΕΝΑ.....	3
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ .....	5
1 ΕΙΣΑΓΩΓΗ.....	7
1.1 Υπόβαθρο και Κίνητρο .....	7
1.2 Στόχοι και Περιορισμοί.....	8
2 ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ.....	9
2.1 Εισαγωγή στην Εκπαιδευτική Ρομποτική.....	9
2.2 Συστήματα Εκπαιδευτικής Ρομποτικής.....	10
2.3 Επίδραση των Συστημάτων Ρομποτικής στην Εκπαίδευση .....	12
3 ΑΝΑΛΥΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΡΟΜΠΟΤ .....	13
3.1 Εκλογή Εκπαιδευτικού Συστήματος.....	13
3.2 Αποσαφήνιση Υπάρχουσας Λειτουργίας.....	14
3.3 Εύρεση Πιθανών Αλλαγών .....	18
3.3.1 Αλλαγή Γλώσσας.....	18
3.3.2 Προσθήκη Ονομασίας Τοπικού Domain .....	18
3.3.3 Ολοκλήρωση Επιπέδου Λειτουργίας «Δύσκολο» .....	19
3.3.4 Επέκταση Υλικών Δυνατοτήτων.....	20
4 ΥΛΟΠΟΙΗΣΗ ΑΝΑΒΑΘΜΙΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	21
4.1 Σχεδιασμός Προσέγγισης.....	21
4.2 1 <sup>ος</sup> Κύκλος Βελτιώσεων Συστήματος .....	22
4.2.1 Αλλαγή Γλώσσας Ιστοσελίδας .....	22
4.2.2 Προσθήκη Ονομασίας Τοπικού Domain .....	23
4.2.3 Προσθήκη ESP32 Πλακέτας & ESP - NOW .....	25
4.3 2 <sup>ος</sup> Κύκλος Βελτιώσεων Συστήματος .....	28
4.3.1 Συνδυασμός Κώδικα Από Edge Impulse .....	29
4.3.2 Επεξεργαστής Γραπτού Κώδικα .....	35
4.3.3 Προσθήκη Turtle Graphics .....	38
4.3.4 Βιβλιοθήκη Skulpt σε JavaScript.....	40
4.3.5 Ολοκλήρωση Ιστοσελίδας .....	43
4.4 Ανεπιτυχής λειτουργία .....	45

5	ΣΥΖΗΤΗΣΗ .....	53
5.1	Συζήτηση.....	53
5.2	Μελλοντική Έρευνα .....	54
5.2.1	Βιβλιοθήκη Wi-Fi Manager .....	54
5.2.2	Σύνδεση με Γλώσσα Scratch .....	55
5.2.3	ESP32 Τηλεχειριστήριο .....	56
5.2.4	Χρήση Bootstrap UI και Bootswatch θεμάτων.....	56
6	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	57
7	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	58
8	ΠΑΡΑΡΤΗΜΑΤΑ .....	59



## ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Τα περιεχόμενα του λειτουργικού συστήματος, όπου (a) ο αρχικός φάκελος με το πρόγραμμα του ESP8266, και (b) το πρόγραμμα της ιστοσελίδας χειρισμού.....	14
Εικόνα 2: Λογικό διάγραμμα λειτουργίας ιστοσελίδας ελέγχου .....	15
Εικόνα 3: Αποσαφήνιση εντολοσειράς πολλαπλών πακέτων εντολών .....	16
Εικόνα 4: Διαδικασία εκκίνησης του συστήματος.....	17
Εικόνα 5: Λογικό διάγραμμα ελέγχου ρομποτικού συστήματος .....	17
Εικόνα 6: Λογικό διάγραμμα ελέγχου ρομποτικού συστήματος .....	19
Εικόνα 7: Στιγμιότυπο οθόνης από Notepad++ και σελίδων προγράμματος.....	23
Εικόνα 8: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης mDNS.....	24
Εικόνα 9: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης ESP - NOW .....	27
Εικόνα 10: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης ESP - NOW σε ESP32.....	28
Εικόνα 11: Στιγμιότυπο οθόνης από Notepad++ και επιλογής AI Thinker ESP32 - CAM στον κώδικα του Edge Impulse .....	30
Εικόνα 12: Στιγμιότυπο οθόνης από Notepad++ και κώδικα του Edge Impulse για εκτύπωση αποτελεσμάτων σε σειριακή θήρα.....	30
Εικόνα 13: Στιγμιότυπο οθόνης από Notepad++ και κώδικα παραδείγματος.....	33
Εικόνα 14: Στιγμιότυπο οθόνης από Notepad++ και κώδικα ESP8266 για συσχέτιση εντολών .....	34
Εικόνα 15: Στιγμιότυπο οθόνης από CodeMirror σε δοκιμαστική ιστοσελίδα .....	36
Εικόνα 16: Στιγμιότυπο οθόνης από CodeMirror με CSS σε δοκιμαστική ιστοσελίδα .....	37
Εικόνα 17: Στιγμιότυπο οθόνης από CodeMirror με CSS Python σε «Δύσκολο» επίπεδο ελέγχου .....	37
Εικόνα 18: Στιγμιότυπο οθόνης από ιστοσελίδα με JavaScript και Turtle Graphics. 39	
Εικόνα 19: Στιγμιότυπο οθόνης από ιστοσελίδα οχήματος με Turtle Graphics .....	40
Εικόνα 20: Λογικό διάγραμμα λειτουργίας σελίδας για «δύσκολο» επίπεδο.....	41
Εικόνα 21: Στιγμιότυπο οθόνης από ιστοσελίδα οχήματος με Turtle Graphics & Skulpt.....	42

Εικόνα 22: Στιγμιότυπο οθόνης από κώδικα ιστοσελίδας με μεταγλώττιση εντολών χρήστη σε εντολές χελώνας .....	43
Εικόνα 23: Στιγμιότυπο οθόνης από κώδικα ιστοσελίδας με μεταγλώττιση εντολών χρήστη σε εντολές οχήματος .....	44
Εικόνα 24: Στιγμιότυπο οθόνης από τελική ιστοσελίδα οχήματος στο «δύσκολο» επίπεδο .....	45
Εικόνα 25: Στιγμιότυπο οθόνης από το γράφημα σύγκυσης του εκπαιδευόμενου μοντέλου (αρχικά αποτελέσματα) .....	46
Εικόνα 26: Ενδεικτικό στιγμιότυπο οθόνης κατά την σχεδίαση μιας από τις κάρτες ελέγχου.....	47
Εικόνα 27: Εικόνες του πάγκου εργασίας κατά την λήψη φωτογραφιών μέσω της πλακέτας ESP32 για το μοντέλο εκπαίδευσης οπτικής αναγνώρισης.....	48
Εικόνα 28: Ιστοσελίδα ελέγχου πλακέτας ESP32 και του προγράμματος <i>26_collect_images.ino</i> του EloquentArduino .....	48
Εικόνα 29: Στιγμιότυπα οθόνης από την επιλογή της κάρτας περιστροφής δεξιόστροφα στην πλατφόρμα Edge Impulse.....	49
Εικόνα 30: Στιγμιότυπο οθόνης από την πλατφόρμα Edge Impulse κατά την προσαρμογή ρυθμίσεων.....	50
Εικόνα 31: Αποτελέσματα δοκιμών του μοντέλου αναγνώρισης με χρήση εξωτερικής κάμερας (A) ή της πλατφόρμας Edge Impulse σε τοπικό υπολογιστή (B).....	52
Εικόνα 32: Στιγμιότυπο οθόνης κατά την εκτέλεση του κώδικα στην πλατφόρμα Edge Impulse σε τοπικό υπολογιστή όπου (A) 0.85 βεβαιότητα κάρτα κίνησης πίσω και (B) 0.58 βεβαιότητα κάρτα κίνησης μπροστά. ....	52
Εικόνα 33: Παράδειγμα γλώσσας προγραμματισμού Scratch.....	55



**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**Τμ. ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ**  
**Π.Μ.Σ. «ΜΗ ΕΠΙΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΚΑΙ**  
**ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΑ ΣΥΣΤΗΜΑΤΑ»**

---

## **1 ΕΙΣΑΓΩΓΗ**

### **1.1 Υπόβαθρο και Κίνητρο**

Η εργασία που ακολουθεί θέτει ως βάση της, την ορθή κατανόηση και εφαρμογή, της εκπαιδευτικής ρομποτικής, ως μέσω εκπαίδευσης μαθητών της πρωτοβάθμιας εκπαίδευσης, σε θέματα που αφορούν τον προγραμματισμό, μέσω του οποίου μπορεί να αναπτυχθεί κριτική σκέψη για την επίλυση προβλημάτων. Καθώς επίσης, μπορεί να αποτελέσει μέσω για την εισαγωγή των παιδιών στην επιστήμη της μηχανικής και στον τρόπο με τον οποίο ο άνθρωπος αλληλεπιδρά με το σύμπαν. Με γνώμονα την διαρκή εξέλιξη της τεχνολογίας και πιο συγκεκριμένα τον κλάδο της ρομποτικής, όπου νέες, αρκετά υποσχόμενες τάσεις κατακλύζουν διαρκώς την αγορά, η εκπαίδευση των μαθητών της πρωτοβάθμιας βαθμίδας σε θέματα που αφορούν τις φυσικές επιστήμες, την τεχνολογία αλλά και πιο συγκεκριμένα την μηχανική και τα μαθηματικά, χρήζουν διαρκούς ανανέωσης.

Το κίνητρο που ώθησε τον συγγραφέα της παρούσας διπλωματικής εργασίας στην επιλογή του θέματος αυτού είναι η ανάπτυξη κέντρων εκπαιδευτικής ρομποτικής και STEM (Science, Technology, Engineering and Mathematics) σε όλη την Ελληνική επικράτεια τα τελευταία χρόνια, μέσω των οποίων καταρτισμένοι εκπαιδευτικοί παραδίδουν μαθήματα σε παιδιά όλων των ηλικιών, χρησιμοποιώντας την εκπαιδευτική ρομποτική ως μέσω διατήρησης του ενδιαφέροντος των μαθητών στα θέματα των μαθημάτων, την επίτευξη καλύτερης κατανόησης και την ενίσχυση της εκπαιδευτικής εμπειρίας για το σύνολο της τάξης.

Πέραν της χρήσης νέων τεχνικών και τεχνολογιών για την εκπαίδευση των μαθητών, η διατήρηση του κόστους κατασκευής και λειτουργίας σε χαμηλά και αποδεκτά επίπεδα, σε συνδυασμό με την εύκολη αναπαραγωγή των συστημάτων εκπαίδευσης από όλα τα εκπαιδευτικά κέντρα, είναι τα κύρια μελήματα που καλείται η έρευνα που ακολουθεί να αντιμετωπίσει κατά την εκλογή των προτάσεων εξέλιξης και βελτίωσης των υπαρχόντων συστημάτων.

## 1.2 Στόχοι και Περιορισμοί

Το εύρος του ερευνητικού τμήματος της παρούσας εργασίας εστιάζεται κυρίως στην κατανόηση ενός υποσχόμενου ρομποτικού συστήματος ανοιχτού κώδικα που θα επιλεγεί κατά την διεξαγωγή της διπλωματικής εργασίας και μετέπειτα στην εξέλιξη του με την χρήση παρόμοιων ηλεκτρονικών συσκευών και προγραμμάτων ανοιχτού κώδικα, βελτιώνοντας ουσιαστικά τον τρόπο λειτουργίας του. Επικεντρώνοντας την εργασία στην ανάλυση του συστήματος και την βελτίωση των λειτουργιών του ρομπότ, η έρευνα στοχεύει να κάνει πιο κατανοητή και προσβάσιμη την λειτουργία του προγράμματος καθώς και τον μετέπειτα επαναπρογραμματισμό του με βάση τις ανάγκες των μαθημάτων που θα παρουσιάζονται μέσω αυτού.

Σε αυτό το σημείο θεωρείται σημαντικό να γίνει αναφορά στους περιορισμούς που συναντώνται κατά την συγγραφή της διπλωματικής εργασίας, αναγνωρίζοντας τους με διαφάνεια για να διατηρηθεί η ακεραιότητα και η αξιοπιστία αυτής. Το σύνολο των περιορισμών που παρουσιάζονται στην εργασία αυτή συνοψίζονται στην λίστα που ακολουθεί.

- *Μεθοδολογία:* Η διπλωματική εργασία βασίζεται στην έρευνα και εφαρμογή δυνατοτήτων υπό την προϋπόθεση να είναι ανοιχτού κώδικα και μικρού κόστους για να μπορεί να δημιουργηθεί από μαθητές σε εργαστηριακό περιβάλλον. Ως εκ τούτου, υπόκειται σε εγγενείς περιορισμούς που σχετίζονται με τις δυνατότητες που μπορούν να επιτευχθούν σε αυτήν την προσέγγιση.
- *Χρήση δεδομένων:* Η διαθεσιμότητα και η ποιότητα των δεδομένων που χρησιμοποιούνται κατά την μελέτη και συνεπώς τα αποτελέσματα που προκύπτουν από αυτή, ενέχουν πιθανούς περιορισμούς οι οποίοι υπό περιπτώσεις δεν μπορούν να αντιμετωπιστούν χωρίς την αύξηση του κόστους. Σε αυτή την περίπτωση θα γίνεται αναφορά των δυνατοτήτων χωρίς να γίνεται εμβάθυνση στον τρόπο πραγματοποίησής τους.
- *Εξωτερικοί παράγοντες:* Εξωτερικοί παράγοντες όπως περιορισμοί σε εργαλεία, ανεπαρκή υπολογιστική ισχύ, χρονικά περιθώρια καθώς και ο ανθρώπινος παράγοντας μπορεί να επηρεάσουν τις προτάσεις και το τελικό αποτέλεσμα που θα προκύψει από την εργασία αυτή.

## 2 ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ

### 2.1 Εισαγωγή στην Εκπαιδευτική Ρομποτική

Η εκπαιδευτική ρομποτική αποτελεί μια μεθοδολογία εκπαίδευσης μαθητών, που χρησιμοποιεί την τεχνολογία για την ενίσχυση της μάθησης και την ανάπτυξη δεξιοτήτων σε πολλούς τομείς. Η προσέγγιση της εκπαιδευτικής ρομποτικής επιτρέπει στους μαθητές να εφαρμόσουν τις θεωρητικές γνώσεις που λαμβάνουν εντός του τυπικού μαθήματος, σε πρακτικά προβλήματα, εξερευνώντας έτσι διάφορες πτυχές της τεχνολογίας.

Παράλληλα, μέσω της λύσης προβλημάτων κατά την διεξαγωγή του μαθήματος, ενθαρρύνεται η συνεργασία, η δημιουργικότητα και η κριτική σκέψη, προετοιμάζοντας τους μαθητές για τις απαιτήσεις του σύγχρονου κόσμου. Μέσω της εκπαιδευτικής ρομποτικής, οι μαθητές αποκτούν τη δυνατότητα να αναπτύξουν πρακτικές δεξιότητες σε πολλούς τομείς, όπως είναι οι Φυσικές Επιστήμες, η Τεχνολογία, η Μηχανική και τα Μαθηματικά, τομείς που αποτελούν μέρος του ακρωνυμίου STEM στην αγγλική γλώσσα («Τι είναι STEM», χ.χ.). Η μέθοδος της εκπαιδευτικής ρομποτικής βασίζεται στις θεωρητικές γνώσεις των μαθημάτων STEM, των οποίων ενδυναμώνει την αποτελεσματικότητα μάθησης μέσω της εισαγωγής ρομποτικών συστημάτων, την χρήση πρακτικών προγραμμάτων και εφαρμογών, για την βελτίωση της μαθησιακής εμπειρίας.

Όπως περιγράφεται σε σχετική μελέτη των Stone-MacDonald, Angi, Wendell, Kristen, Douglass, Anne, & Love, Mary Lu, (2015), η επιλογή της εκπαιδευτικής ρομποτικής ως τρόπος εκμάθησης μέσω επίλυσης προβλημάτων από παιδιά, σε συνδυασμό με την μοντελοποίηση προβλημάτων – ασκήσεων βάση του τρόπου σκέψης των ενηλίκων, είναι κρίσιμης σημασίας. Η καθοδήγηση και η υποστήριξη του εκπαιδευτικού προσωπικού κατά την διεξαγωγή των εκπαιδευτικών δραστηριοτήτων, διευκολύνει την καλλιέργεια της περιέργειας, της επιμονής, της ευελιξίας και της συνεργατικότητας στα παιδιά. Μέσα από αυτήν την ενεργή συμμετοχή των ενηλίκων, τα παιδιά παρατηρούν, συμμετέχουν και αποκτούν τις δεξιότητες που απαιτούνται για την ανάπτυξη κριτικής σκέψης και της λύσης προβλημάτων, ενώ μαθαίνουν νέα πράγματα για τον τρόπο με τον οποίο λειτουργεί ο κόσμος γύρω τους.

## 2.2 Συστήματα Εκπαιδευτικής Ρομποτικής

Εμβαθύνοντας σε αυτήν την μέθοδο εκπαίδευσης, ένα πλήθος από διαφορετικά ρομποτικά συστήματα και πλατφόρμες εκπαιδευτικής ρομποτικής κάνουν την εμφάνιση τους, καθένα με τα δικά του χαρακτηριστικά και πλεονεκτήματα. Τα συστήματα αυτά έχουν κατασκευαστεί με συγκεκριμένα λογισμικά προγράμματα, στοχευμένα με τέτοιο τρόπο, που να επιτρέπει στους μαθητές να επαναπρογραμματίζουν τις διάφορες λειτουργίες που το καθένα μπορεί να εκτελεί.

Από τις πιο διαδεδομένες επιλογές ρομποτικών συστημάτων στον χώρο της δευτεροβάθμιας εκπαίδευσης σε εργαστηριακό επίπεδο είναι η χρήση ρομποτικών βραχιόνων, των οποίων οι κινήσεις μπορούν να προγραμματιστούν μέσω εξωτερικής πλακέτας που τα συνοδεύει. Ενώ, στην περίπτωση της πρωτοβάθμιας εκπαίδευσης, προτιμώνται μικρής κλίμακας ρομποτικά συστήματα που είναι πιο προσιτά σε παιδιά όλων των ηλικιών και επιτρέπουν λειτουργίες με βάση τους στόχους της διδασκαλίας. Τα ρομποτικά συστήματα μικρής κλίμακας συχνά εμπεριέχουν διασυνδέσεις με εκπαιδευτικές πλατφόρμες λογισμικού που περιλαμβάνουν ειδικές εφαρμογές για τον προγραμματισμό των ρομποτικών συστημάτων σε έτοιμα παραδείγματα κώδικα.

Ένας ακόμα τρόπος εκπαίδευσης μέσω της ρομποτικής είναι η χρήση ειδικών πακέτων εκπαιδευτικού υλικού, τα οποία περιλαμβάνουν υλικό αλλά και οδηγίες για την κατασκευή και τον προγραμματισμό του συστήματος για ένα πλήθος μαθημάτων και λειτουργιών. Μέσω της πρακτικής εφαρμογής των οδηγιών, οι μαθητές έχουν την δυνατότητα να αναπτύξουν δεξιότητες προγραμματισμού, να μάθουν νέες τεχνολογίες, όσο δοκιμάζουν και πειραματίζονται με την λειτουργία του εκάστοτε συστήματος.

Σε μια προσπάθεια εύρεσης και ανάλυσης των διαφόρων πλεονεκτημάτων, των δημοφιλέστερων επιλογών εκπαιδευτικής ρομποτικής για την εκμάθηση STEM, έγινε επιλογή των συστημάτων που ακολουθούν, για την άντληση πληροφοριών και ιδεών που θα μπορούσαν να χρησιμοποιηθούν στην βελτίωση ενός συστήματος της επιλογής του συγγραφέα. Η σειρά με την οποία επιλέχθηκε να παρουσιαστούν τα ακόλουθα συστήματα είναι βασισμένη από τις πιο ολοκληρωμένες και πιθανότατα κλειστού λειτουργικού συστήματος επιλογών, σε πιο ανοιχτού τύπου επιλογές.

### LEGO Education

Από τις δημοφιλέστερες, αν όχι η πιο δημοφιλής επιλογή σε έτοιμα σεν υλικού για μαθήματα εκπαιδευτικής ρομποτικής, δε θα μπορούσε να είναι άλλο παρά τα εκπαιδευτικά κιτ ρομποτικής της εταιρίας LEGO. Τα πιο γνωστά σεν είναι αυτά της σειράς LEGO Mindstorm και της σειράς LEGO WeDo, τα οποία είναι σχεδιασμένα για να εισάγουν τους μαθητές στον προγραμματισμό και την μηχανική, χρησιμοποιώντας τουβλάκια LEGO και διάφορα ηλεκτρομηχανολογικά εξαρτήματα που φέρουν επιλογές μηχανικής στις κατασκευές των μαθητών. Αν και η τιμή τους είναι αρκετά υψηλή σε σχέση με άλλες επιλογές που ακολουθούν, ο τρόπος κατασκευής και ελέγχου των κατασκευών που μπορεί ο εκάστοτε μαθητής να δημιουργήσει, τα κάνουν αρκετά δελεαστική επιλογή για τα εργαστηριακά κέντρα εκπαίδευσης STEM.

### VEX Robotics

Μια εξίσου δημοφιλής επιλογή σε έτοιμα σεν υλικού είναι αυτή της εταιρίας VEX Robotics. Τόσο η VEX όσο και η LEGO, προσφέρουν το απαραίτητο λογισμικό για τον προγραμματισμό του ρομποτικού συστήματος, καθώς επίσης τα απαραίτητα εργαλεία και πλήθος επιλογών ρομποτικών εξαρτημάτων για την κατασκευή του συστήματος από τους μαθητές.

### Arduino / Raspberry Pi

Διευρύνοντας την λίστα επιλογών σε ρομποτικά συστήματα που βασίζονται σε πλατφόρμες υλικού και λογισμικού ανοιχτού κώδικα, δύο πλατφόρμες αυτού του τύπου μπορούν να σταθούν αντάξιες των ολοκληρωμένων κιτ επιλογών. Το Arduino είναι μια πλατφόρμα ανοιχτού κώδικα που εμπεριέχει ένα πλήθος από πλακέτες οι οποίες μπορούν να χρησιμοποιηθούν ως βάση για την δημιουργία ρομποτικών συστημάτων. Ο προγραμματισμός αυτών γίνεται σε γλώσσα προγραμματισμού Arduino, μια ειδικά κατασκευασμένη γλώσσα που είναι αποδεκτή από πολλών ειδών πλακέτες ανοιχτού κώδικα.

Παράλληλα, υπάρχει η επιλογή του Raspberry Pi το οποίο βασίζεται επίσης σε πλακέτα ανοιχτού κώδικα όπως το Arduino, ωστόσο ομοιάζει περισσότερο σε ένα μικρό υπολογιστή με περιορισμένες δυνατότητες επεξεργαστικής ισχύος. Και οι δύο επιλογές μπορούν να αποτελέσουν την βάση για την δημιουργία προγραμμάτων

αλλά και συστημάτων εκπαιδευτικής ρομποτικής ανοιχτού κώδικα, ενώ υπάρχουν πολλές κατασκευάστριες εταιρίες οι οποίες κατασκευάζουν πλακέτες ανοιχτού κώδικα με βάση αυτές. Μερικές από αυτές που αξίζει να αναφερθούν σε αυτό το σημείο είναι η πλακέτα ESP8266, ESP32, Raspberry Pi Pico, Nvidia Jetson και STM32. Αυτές οι πλακέτες φέρουν πιο ευρεία χρήση που ξεπερνάει τα όρια της εκπαιδευτικής ρομποτικής, αλλά υπό περιπτώσεις είναι αρκετά καλές επιλογές σε σχέση με τις τυπικές πλακέτες Arduino και Raspberry Pi.

### Scratch / Code.org

Εκτός από τις πλατφόρμες που εστιάζουν σε έτοιμες ηλεκτρονικές πλακέτες για την δημιουργία ρομποτικών συστημάτων, υπάρχουν και πλατφόρμες που εστιάζουν στο κομμάτι του προγραμματισμού και της ρομποτικής από την σκοπιά της πληροφορικής. Τέτοιες πλατφόρμες είναι η Scratch και το Code.org, που παρέχουν περιβάλλον προγραμματισμού που διευκολύνει της εισαγωγή των παιδιών στον κόσμο της πληροφορικής και της δημιουργίας κώδικα μέσω διαδραστικών παιχνιδιών και εφαρμογών που μπορούν να συνδυαστούν με την ρομποτική.

## **2.3 Επίδραση των Συστημάτων Ρομποτικής στην Εκπαίδευση**

Αναζητώντας την επίδραση που φέρει η χρήση συστημάτων ρομποτικής στην εκπαίδευση των μαθητών και ειδικά σε νεαρή ηλικία, αρκεί κανείς να αναλογιστεί πως άλλα υλικά συστήματα χρησιμοποιούνται ήδη για εκπαιδευτικούς σκοπούς και τι αντίκτυπο επιφέρουν. Ένα μεγάλο παράδειγμα υλικού συστήματος που χρησιμοποιείται στην προσχολική εκπαίδευση των μαθητών, είναι τα γνωστά σε όλους τουβλάκια, που στόχο έχουν να προωθήσουν τη δημιουργικότητα, την αντοχή και τη συνεργατική εργασία μεταξύ παιδιών.

Μέσω των παρατηρήσεων που προκύπτουν κατά την διεξαγωγή κατασκευών με τουβλάκια, οι μαθητές μπορούν να εξάγουν συμπεράσματα τα οποία συσχετίζουν το παιχνίδι με πρακτικές γνώσεις όσον αφορά την μηχανική και την φυσική (Moosaw, S., 2012). Κατά αντιστοιχία, η χρήση της εκπαιδευτικής ρομποτικής είναι το επόμενο βήμα στην κατανόηση των μαθημάτων STEM, ιδιαίτερα κατά την πρωτοβάθμια εκπαίδευση όπου οι μαθητές δεν εμβαθύνουν σε πολλαπλά επιστημονικά παιδιά, όπως γίνεται στην δευτεροβάθμια εκπαίδευση.



### 3 ΑΝΑΛΥΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΡΟΜΠΟΤ

#### 3.1 Εκλογή Εκπαιδευτικού Συστήματος

Καθημερινά η εκπαιδευτική κοινότητα αντιμετωπίζει νέες προκλήσεις αλλά και ευκαιρίες, καθώς η τεχνολογία παίρνει ολοένα και μεγαλύτερο ρόλο στην ζωή του ανθρώπου, με την ενσωμάτωση της τεχνολογίας στην διαδικασία της εκπαίδευσης να τείνει, να γίνει αναπόφευκτη. Στο πλαίσιο αυτό, η παρούσα διπλωματική εργασία εξετάζει την επιλογή ενός εκπαιδευτικού ρομποτικού συστήματος για την πρωτοβάθμια εκπαίδευση, με έμφαση στους τομείς που καλύπτει το STEM.

Η τελική εκλογή του εκπαιδευτικού ρομποτικού συστήματος βασίστηκε στην χρήση λειτουργικού ανοιχτού κώδικα, δίνοντας έμφαση στην ευελιξία και την δυνατότητα προσαρμογής, κρατώντας χαμηλό το κόστος δημιουργίας. Αν και η επιλογή είναι λιγότερο δημοφιλής σε σχέση με προκαθορισμένες λύσεις κλειστών λογισμικών από μεγάλες και γνωστές εταιρίες, η εργασία βασίζεται στην δυνατότητα βελτίωσης και προσαρμογής του εκάστοτε συστήματος στα μέτρα και τις ανάγκες του μαθήματος στο οποίο προορίζεται να χρησιμοποιηθεί.

Το σύστημα που επιλέχθηκε είναι αυτό ενός μικρού μεγέθους τηλεκατευθυνόμενου οχήματος το οποίο πραγματοποιήθηκε τα τελευταία χρόνια από τον επιβλέποντα καθηγητή της παρούσας διπλωματικής σε πλαίσια δικής του έρευνας. Αυτό είχε ως αποτέλεσμα την εύκολη πρόσβαση σε πληροφορίες αλλά και στον κώδικα του οχήματος, την καλύτερη ανταλλαγή απόψεων και γνώσεων, ενώ παράλληλα μέσω του οράματος του καθηγητή για το σύστημα που δημιούργησε, μπόρεσε ο συγγραφέας να ορίσει τους δικούς του στόχους για τις βελτιώσεις που θα μπορούσαν να πραγματοποιηθούν.

Στα ιδιαίτερα χαρακτηριστικά που ακολουθούν το σύστημα που επιλέχθηκε είναι η κατασκευή του σκελετού του οχήματος, με τεχνολογία 3D printing, προσφέροντας πληθώρα έτοιμων επιλογών για τον σχεδιασμό και την προσαρμογή του κελύφους του οχήματος στις όποιες ανάγκες θα έχει το μάθημα στο οποίο θα χρησιμοποιηθεί. Η ευελιξία αυτή, συνοδευόμενη με τον ανοιχτού τύπου λογισμικό, όπως παρουσιάζεται μέσω του ακαδημαϊκού άρθρου των Chatzopoulos, Avraam & Kalogiannakis, Michail & Papadakis, Stamatios & Papoutsidakis, Michail (2022),

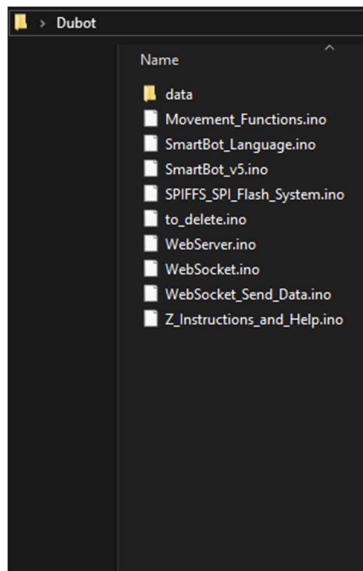
επιτρέπει την δημιουργία σχεδιασμών και την προσθήκη εξαρτημάτων, με στόχο το όχημα να ομοιάζει λ.χ. μια πασχαλίτσα ή μια μέλισσα, προκειμένου να ενθαρρύνεται η αλληλεπίδραση μεταξύ του ρομπότ και των παιδιών.

### 3.2 Αποσαφήνιση Υπάρχουσας Λειτουργίας

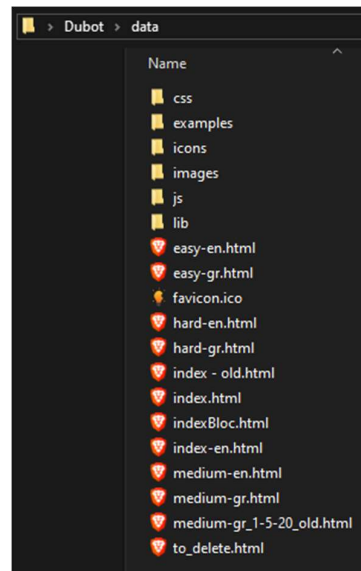
Το σύνολο του ανοιχτού κώδικα που συνοδεύει το επιλεγμένο ρομποτικό σύστημα, παραδόθηκε στον συγγραφέα της διπλωματικής εργασίας από τον δημιουργό του προγράμματος κ. Χατζόπουλο Αβραάμ, υπό την μορφή συμπιεσμένου αρχείου (ZIP file).

Όπως φαίνεται στην Εικόνα 1 που ακολουθεί, το περιεχόμενο του φακέλου που εμπεριέχει όλα τα αρχεία για το λειτουργικό σύστημα (a) αποτελείται από δύο κύρια μέρη, τον βασικό κώδικα για την πλακέτα ESP8266 και το σύνολο των δεδομένων που χρειάζεται για να δημιουργήσει και να ελέγξει το όχημα μέσω της ιστοσελίδας.

Ο φάκελος με τα δεδομένα (data) (b) παρουσιάζει επίσης ενδιαφέρον καθώς χωρίζει τις υποκατηγορίες δεδομένων σε φακέλους, ενώ ο κώδικας για την μορφοποίηση της ιστοσελίδας βρίσκεται διαθέσιμος ως έχει εντός του φακέλου αυτού.



(a)



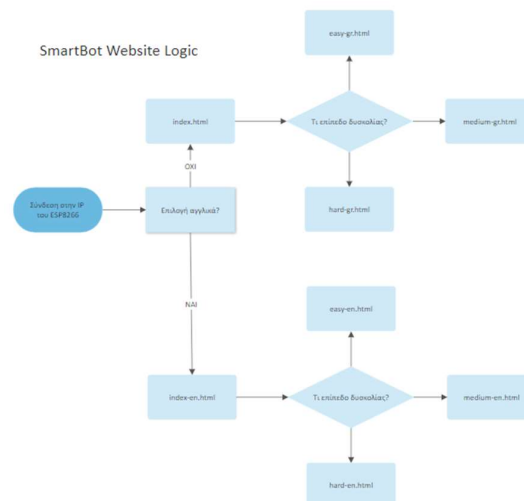
(b)

Εικόνα 1: Τα περιεχόμενα του λειτουργικού συστήματος, όπου (a) ο αρχικός φάκελος με το πρόγραμμα του ESP8266, και (b) το πρόγραμμα της ιστοσελίδας χειρισμού

Στο σύνολο τους, τα αρχεία με κατάληξη «.ino» είναι αρχεία κώδικα που σχετίζονται με την χρήση γλώσσας προγραμματισμού Arduino, μια γλώσσα προγραμματισμού η οποία βασίζεται στον τρόπο με τον οποίο η πλακέτα είναι συνδεδεμένη στα επιμέρους υποσυστήματα, μέσω καλωδίωσης, και ομοιάζει αρκετά την γλώσσα προγραμματισμού C/C++. Ο διαχωρισμός του κώδικα σε πολλαπλά αρχεία, έγινε από τον δημιουργό του προγράμματος πιθανότατα για την διευκόλυνση στην μετέπειτα εύρεση και επεξεργασία συγκεκριμένων λειτουργιών του συστήματος.

Παράλληλα, τα αρχεία που βρίσκονται εντός του φακέλου data έχουν καταλήξεις όπως «.html» και «.css», που αντιστοιχούν σε κώδικα σχεδιασμού και διαμόρφωσης ιστοσελίδων. Ενώ, αρχεία με κατάληξη «.js» σχετίζονται με την γλώσσα προγραμματισμού JavaScript και τα «.png» σε εικονίδια που χρησιμοποιούνται για διάφορους λόγους στο πρόγραμμα.

Εμβαθύνοντας περισσότερο στον φάκελο data και πιο συγκεκριμένα στα αρχεία με κατάληξη «.html», γίνεται εμφανές ο τρόπος χειρισμού του συστήματος μέσω ιστοσελίδων, που λειτουργούν ως δίαυλοι επικοινωνίας, ελέγχοντας κάθε διαθέσιμη κίνηση του οχήματος. Διαβάζοντας τα ονόματα των αρχείων και μετέπειτα συσχετίζοντας τα μέσω του κώδικα που περιέχει το κάθε ένα από αυτά, παρατηρείται ο τρόπος με τον οποίο συνδέονται μεταξύ τους όλες οι σελίδες που παρουσιάζονται στην Εικόνα 1 (b). Ενώ στην Εικόνα 2, παρουσιάζεται η λογική πίσω από το σύνολο των σελίδων και των μεταξύ τους σχέσεων.



Εικόνα 2: Λογικό διάγραμμα λειτουργίας ιστοσελίδας ελέγχου

Οι εντολές χειρισμού που αποστέλλονται μέσω της ιστοσελίδας στην πλακέτα ελέγχου φαίνεται να φέρουν μια τυποποιημένη μορφή πακέτου, το οποίο και αναλύεται στην Εικόνα 3. Η αρχική παρατήρηση γίνεται μέσω της κονσόλας του φυλλομετρητή που επιλέχθηκε από τον συγγραφέα της διπλωματικής για το άνοιγμα της ιστοσελίδας, ενώ στην συνέχεια μέσω ελέγχου στα επί μέρους αρχεία της πλακέτας, όπου βρίσκονται και οι αντίστοιχες προγραμματισμένες λειτουργίες που τυποποιούν κάθε εντολή του χειριστή.



Εικόνα 3: Αποσαφήνιση εντολοσειράς πολλαπλών πακέτων εντολών

Όταν ο χειρισμός γίνεται μέσω του «εύκολου» επιπέδου δυσκολίας, η εντολή έχει την μορφή μονού πακέτου εντολής (λ.χ. [\"όνομα εντολής\", τιμή εντολής]), ενώ από το «μεσαίο» επίπεδο δυσκολίας και άνω, όπου πολλαπλές εντολές μπορεί να σταλούν μαζικά, το πακέτο φέρει μορφή εντολοσειράς (λ.χ. [εντολή\_1, εντολή\_2, εντολή\_3]), όπου πολλά πακέτα εντολών, διαχωρίζονται με κόμμα εντός τετραγωνισμένης παρενθέσεως.

Παράλληλα, ενδιαφέρον φέρει το *file.js* με την χρήση της λειτουργίας *runRobot* και του *connection.send(datatosend);*, όπου το *datatosend* είναι η συμβολοσειρά που περιέχει τις παραπάνω εντολές. Επίσης, μέσω της γραμμής 12 του κώδικα, όπου ορίζεται η μεταβλητή *datatosend*:

```
Var datatosend = JSON.stringify(blocks.map(block.script));
```

παρατηρείται ότι ισοδυναμεί με μια σειρά αντικειμένων τα οποία έχουν συνδεθεί σε συνδυαστικές λειτουργίες όπου κάθε αντικείμενο έχει τον δικό του κώδικα. Μέσω της μετατροπής αυτής σε κώδικα κάθε αντικειμένου, είναι δυνατή η μετάφραση –

αποκωδικοποίηση των μετακινούμενων εντολών της «μεσαίας» δυσκολίας σε εντολές που μπορεί να χρησιμοποιήσει το σύστημα. Στο αρχείο `socket.js` οι λειτουργίες αυτές φαίνονται πιο καθαρά μέσω των `sendData(data)` και `sendMovement(data)` λειτουργιών, όπου `data` είναι οι τελικές εντολές για το όχημα. Η Εικόνα 4 παρουσιάζει μέσω λογικού διαγράμματος την σειρά που ακολουθεί το πρόγραμμα από την εκκίνηση του, έως το σημείο όπου μπορεί να λάβει τα πακέτα εντολών από τον χειριστή.

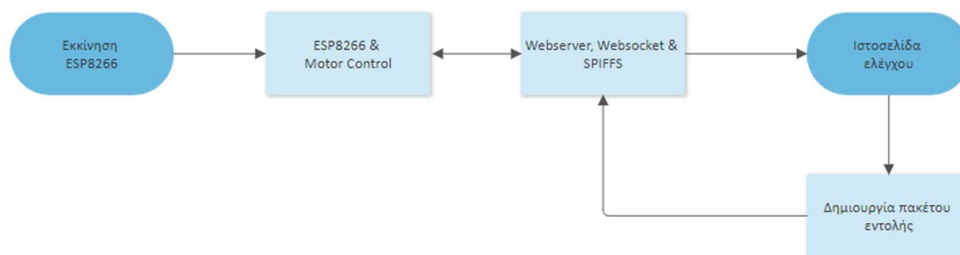
#### SmartBot Operating Architecture



Εικόνα 4: Διαδικασία εκκίνησης του συστήματος

Παράλληλα, μέσω της Εικόνα 5 το λογικό διάγραμμα μεταβάλλεται έτσι ώστε να γίνεται πιο αντιληπτός, τόσο οι σχέσεις μεταξύ τμημάτων του συστήματος όσο και του τρόπου με τον οποίο τα παραπάνω αναφερόμενα πακέτα επηρεάζουν τελικά το ρομποτικό σύστημα.

#### SmartBot Controlling Architecture



Εικόνα 5: Λογικό διάγραμμα ελέγχου ρομποτικού συστήματος

Ο Arduino κώδικας που χρησιμοποιείται για την πλακέτα ESP8266 είναι διαχωρισμένος σε ένα πλήθος αρχείων, μερικά από τα οποία αξίζει να σημειωθούν να είναι τα εξής: `WebSocket.ino`, `WebServer.ino`, `SmartBot_v5.ino` και `Movement_Functions.ino`.

Αν και όλα τα αρχεία έχουν σημασία για την ομαλή λειτουργία του συστήματος, τα προαναφερθέντα αρχεία φέρουν τις πιο σημαντικές πτυχές του κώδικα, καθώς τα δύο πρώτα ευθύνονται για την δημιουργία διακομιστή (server) στον οποίο θα βρίσκεται η ιστοσελίδα διαχείρισης του οχήματος, ενώ τα δύο τελευταία ευθύνονται για την ορθή εκκίνηση της πλακέτας και την μετατροπή των εντολών σε φυσικές κινήσεις του οχήματος.

### **3.3 Εύρεση Πιθανών Αλλαγών**

Έχοντας βρει το εκπαιδευτικό ρομποτικό σύστημα στο οποίο θα βασιστεί η διπλωματική εργασία και εφόσον έχει αποσαφηνιστεί ο τρόπος με τον οποίο λειτουργεί κάθε υποσύστημα που περιλαμβάνεται σε αυτό, θα πρέπει να βρεθούν και να αναλυθούν οι πιθανές αλλαγές που θα μπορούσαν να γίνουν σε αυτό, στοχεύοντας στην καλύτερη χρήση του από τους μαθητές της πρωτοβάθμιας εκπαίδευσης.

#### **3.3.1 Αλλαγή Γλώσσας**

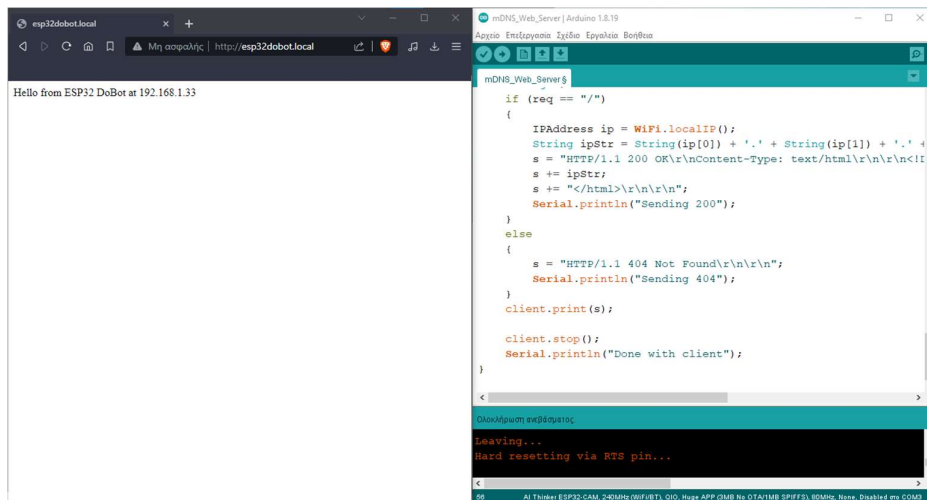
Το πρόγραμμα που ακολουθεί το ρομποτικό σύστημα πέραν από πλήκτρα ελέγχου, εμπεριέχει ονόματα, περιγραφές και ρυθμίσεις που ως επί το πλείστον είναι βασισμένες στην ελληνική γλώσσα. Παράλληλα, από τον δημιουργό του ρομποτικού οχήματος υπάρχει πρόβλεψη για την επιλογή της αγγλικής γλώσσας για το σύνολο της διεπαφής ελέγχου (control interface) η οποία ωστόσο δεν έχει εφαρμοστεί. Μια πιθανή αλλαγή – προσθήκη στο υπάρχον πρόγραμμα είναι η μετάφραση της ιστοσελίδας στην οποία υπάρχει η διεπαφή ελέγχου, στην αγγλική γλώσσα, προβλέποντας έτσι την διευκόλυνση χρήσης του προγράμματος από άτομα που ίσως να μην γνωρίζουν καλά την Ελληνική γλώσσα, ή ακόμα και από μαθητές του εξωτερικού.

#### **3.3.2 Προσθήκη Ονομασίας Τοπικού Domain**

Η ιστοσελίδα που δημιουργείται από το σύστημα μέσω της οποίας γίνεται ο έλεγχος του οχήματος, φέρει ως αρχικό όνομα την διεύθυνση διαδικτυακού πρωτοκόλλου (IP Address) της συσκευής ελέγχου. Συνεπώς, για την επίτευξη σύνδεσης μέσω ενός φυλλομετρητή (Browser) στην κεντρική ιστοσελίδα ελέγχου, πέραν της σύνδεσης στο τοπικό δίκτυο που δημιουργεί το ESP8266, ο χρήστης θα πρέπει είτε να γνωρίζει

εκ των προτέρων την διεύθυνση που λαμβάνει για να συνδεθεί σε αυτό, είτε να είναι συνδεδεμένος σειριακά με το ESP8266 ώστε να δει μέσω της κονσόλας την διεύθυνση που λαμβάνει.

Σε αυτήν την περίπτωση μια πιθανή βελτίωση θα μπορούσε να είναι η χρήση multicast DNS (mDNS) πρωτοκόλλου το οποίο είναι συμβατό σχεδόν με όλες τις μοντέρνες συσκευές που έχουν δυνατότητα σύνδεσης στο διαδίκτυο, κατά το οποίο γίνεται σύνδεση ονομάτων ιστοσελίδων σε διευθύνσεις IP εντός μικρών, τοπικών δικτύων που δεν περιλαμβάνουν διακομιστές ονομάτων (DNS) όπως στην περίπτωση του ESP8266. Για επιβεβαίωση της παρούσας ιδέας, έγινε χρήση πρόχειρου κώδικα σε μια διαθέσιμη πλακέτα ESP32 του συγγραφέα η οποία είναι παρόμοια με το ESP8266, με μόνη διαφορά την χρήση διαφορετικών βιβλιοθηκών.



Εικόνα 6: Λογικό διάγραμμα ελέγχου ρομποτικού συστήματος

Έτσι, όπως φαίνεται και στην Εικόνα 6, οι χρήστες θα μπορούν πλέον να καλούν την ιστοσελίδα με βάση το όνομα της (λ.χ. <http://esp32dobot.local>), χωρίς την ανάγκη της διεύθυνση IP της συσκευής (λ.χ. 192.168.1.33) αν και είναι εφικτή η σύνδεση καλώντας την IP μέσω του φυλλομετρητή, εάν είναι γνωστή.

### 3.3.3 Ολοκλήρωση Επιπέδου Λειτουργίας «Δύσκολο»

Η αρχική ιστοσελίδα που συνδέει τον χρήστη με τον έλεγχο του οχήματος, παρέχει την δυνατότητα επιλογής του επιπέδου δυσκολίας στο οποίο θα αντιστοιχούν και τα πλήκτρα χειρισμού. Έτσι, στο «εύκολο» επίπεδο χειρισμού του συστήματος, η

διεπαφή ελέγχου εμπεριέχει βασικά πλήκτρα που αντιστοιχούν στις κινήσεις που μπορεί να κάνει το όχημα, ενώ στο «μέτριο» επίπεδο δυσκολίας, τα πλήκτρα αντικαθίστανται από πλακίδια με εντολές, τα οποία ο χειριστής μπορεί να θέσει στην γραμμή εντολών με σκοπό την δημιουργία μιας ακολουθίας κινήσεων για το σύστημα.

Αντίστοιχα, έχει γίνει πρόβλεψη για την εισαγωγή και τρίτης βαθμίδας δυσκολίας η οποία ωστόσο δεν έχει υλοποιηθεί ακόμα. Μια πιθανή βελτίωση – εξέλιξη για το πρόγραμμα, είναι η ολοκλήρωση αυτού του επιπέδου δυσκολίας βάση των δυνατοτήτων που θα μπορούσαν να προστεθούν, ακολουθώντας τις γενικές οδηγίες που αναφέρονται στο άρθρο του Χατζόπουλου κ. συν. (2022) για πιθανή προσθήκη προηγμένων εντολών και δυνατότητα προεπισκόπησης της κίνησης του προγράμματος μέσω της σελίδας.

### **3.3.4 Επέκταση Υλικών Δυνατοτήτων**

Η πλακέτα που χρησιμοποιείται για την δημιουργία του ρομποτικού συστήματος έχει περιορισμένο αριθμό εισόδων – εξόδων, και σχετικά μικρής δυνατότητας επεξεργαστική ισχύ, δεδομένα που σχετίζονται άμεσα με το μικρό κόστος της. Η προσθήκη επεκτάσεων όπως μικροφώνου για ηχητικές εντολές, ηχείων για την παραγωγή ήχου κατά την λειτουργία του οχήματος, κάμερας ή άλλων αισθητήρων όπως συνήθως συνδυάζονται σε αρκετά πιο ακριβά, έτοιμα συστήματα ρομποτικής εκπαίδευσης μπορεί να αποδειχτεί αρκετά δύσκολη υπό συνθήκες, αν όχι ακατόρθωτη.

Μια πιθανή λύση που μπορεί να βελτιώσει ως ένα βαθμό τις δυνατότητες του προγράμματος αλλά και την επεκτασιμότητα του από πλευράς υλικού, είναι ο συνδυασμός δευτερεύοντος επεξεργαστή μέσω επιπλέον πλακέτας ή αλλαγή της υπάρχουσας προεπιλεγμένης πλακέτας με κάποια που έχει πιο μεγάλη επεξεργαστική ισχύ, που θα προσθέτει δυνατότητες στο βαθμό του θεμιτού χωρίς να ξεφεύγει από τον όποιο οικονομικό περιορισμό υπάρχει. Τόσο ο τρόπος διασύνδεσης όσο και η επιλογή δεύτερης πλακέτας, αποτελεί επιλογή του κάθε κατασκευαστή, από ένα πλήθος διαθέσιμων επιλογών.



## 4 ΥΛΟΠΟΙΗΣΗ ΑΝΑΒΑΘΜΙΣΗΣ ΣΥΣΤΗΜΑΤΟΣ

### 4.1 Σχεδιασμός Προσέγγισης

Χρησιμοποιώντας τον πηγαίο κώδικα του Χατζόπουλου κ. συν. (2022) ως βάση για την υλοποίηση αναβαθμίσεων στο ρομποτικό σύστημα, σε συνεργασία με τις πληροφορίες που συλλέχθηκαν στο προηγούμενο κεφάλαιο, πραγματοποιούνται κύκλοι συλλογής ιδεών για όλες τις πιθανές λειτουργίες και στην συνέχεια ανάλυση αυτών που θα μπορούσαν να προστεθούν στο σύστημα αναφοράς.

Πίνακας 1: Πίνακας υποψήφιων αναβαθμίσεων για το ρομποτικό όχημα

Πίνακας Υποψήφιων Αναβαθμίσεων Για Όχημα			Έκδοση 2
A / E	Αναβαθμίσεις	Περιγραφή	
A	Μετάφραση	Μετάφραση όλων των σελίδων στα Αγγλικά.	
A	Domain	Χρήση mDNS πρωτοκόλλου για ονομασία σελίδας.	
E	Wi-Fi Manager	Χρήση βιβλιοθήκης εντολών Wi-Fi Manager για τον έλεγχο σύνδεσης σε προϋπάρχοντα δίκτυα.	
E	Επεξεργαστής Γραπτού κώδικα CodeMirror	Προσθήκη λειτουργίας συγγραφής γραπτού κώδικα μέσω της ιστοσελίδας για τον έλεγχο του οχήματος.	
A	Γραφικά Χελώνας	Χρήση γραφικών χελώνας της Python για την προσομοίωση κίνησης του οχήματος στην ιστοσελίδα.	
E	Μεταγλωττιστής Skulpt	Χρήση μεταγλωττιστή στον διάυλο ελέγχου του οχήματος για τον προγραμματισμό κινήσεων με την χρήση Python.	
E	Scratch Plugin	Σύνδεση του προγράμματος ελέγχου του οχήματος με την γλώσσα προγραμματισμού Scratch.	
A	ESP - NOW	Προσθήκη επικοινωνίας μέσω ESP – NOW πρωτοκόλλου για σύνδεση μεταξύ πλακετών Espressif.	
E	Edge Impulse	Συνδυασμός προγράμματος ελέγχου του οχήματος με την πλατφόρμα μηχανικής μάθησης Edge Impulse.	
E	Έλεγχος με κάρτες / σήματα	Συνδυασμός οπτικής αναγνώρισης αντικειμένων με τον έλεγχο του οχήματος.	
A	Πλακέτα ESP32	Αντικατάσταση ή προσθήκη δευτερεύουσας πλακέτας για επέκταση δυνατοτήτων.	
E	Τηλεχειριστήριο	Σχεδιασμός και εφαρμογή τηλεχειριστηρίου για τον έλεγχο του οχήματος.	

Τα στοιχεία που προκύπτουν από την συλλογή ιδεών προστίθενται στον πίνακα 1 με σκοπό την συγκέντρωσή τους, διαχωρίζοντας τα σε (Α) Απαραίτητες αναβαθμίσεις και (Ε) Επιθυμητές αναβαθμίσεις. Στον πίνακα αυτό γίνεται η διασταύρωση των ιδεών και η περιγραφή τους, ώστε να αποφευχθούν τυχόν επαναλήψεις ή ιδέες που δεν θα μπορούσαν να εφαρμοστούν για διάφορους λόγους. Με την ολοκλήρωση των κύκλων ερευνών και την συλλογή ιδεών για τις απαραίτητες διαμορφώσεις που θα πρέπει να γίνουν, ακολουθεί ο διαχωρισμός και η σταδιακή προσθήκη όλων των επιμέρους ιδεών στον πηγαίο κώδικα του συστήματος αναφοράς.

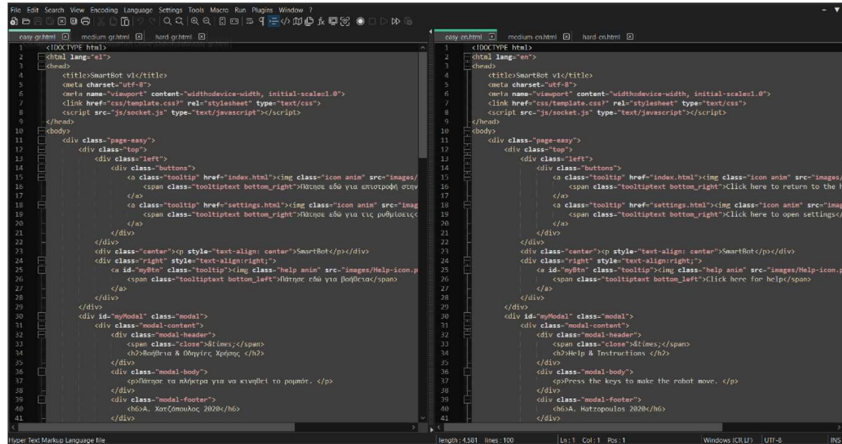
#### **4.2 1<sup>ος</sup> Κύκλος Βελτιώσεων Συστήματος**

Εμβαθύνοντας σε πιο τεχνικά χαρακτηριστικά, οι αναβαθμίσεις που έχουν επιλεγεί στον πρώτο κύκλο έρευνας και συλλογής ιδεών, διαχωρίζονται σε αυτοτελή υποενότητες που περιγράφουν τόσο την ιδέα όσο και την εκτέλεση της προσθήκης στο υπάρχον σύστημα, καταγράφοντας το εκάστοτε τελικό αποτέλεσμα.

##### **4.2.1 Αλλαγή Γλώσσας Ιστοσελίδας**

Δίνοντας μεγάλη βάση στην προσβασιμότητα του εν λόγω ρομποτικού συστήματος για την εκπαιδευτική ρομποτική, η προσθήκη της αγγλικής γλώσσας σαν επιλογή και αντίστοιχα η ολοκλήρωση της μετάφρασης των επί μέρους ιστοσελίδων, ορίστηκε ως απαραίτητη αναβάθμιση και ως το πρώτο βήμα στην βελτίωση του οχήματος. Η αλλαγή αυτή είναι εφικτή με την επεξεργασία του HTML κώδικα του προγράμματος που προορίζεται να εμφανίζεται όταν καλείται η αγγλική έκδοση της σελίδας.

Με την χρήση του προγράμματος Notepad++ (πρόγραμμα επεξεργασίας αρχείων) και του πηγαίου κώδικα που παραδόθηκε στον συγγραφέα από την ομάδα του ρομποτικού οχήματος (Χατζόπουλος κ. συν.), έγινε συλλογή των αρχείων κώδικα που εμπεριέχουν αυτούσιες τις ιστοσελίδες του οχήματος στην ελληνική γλώσσα και στην συνέχεια, έγινε μετάφραση του περιεχομένου τους στην Αγγλική γλώσσα. Στο σύνολο του έργου υπάρχουν 3 ιστοσελίδες που θα πρέπει να μεταφραστούν στην αγγλική γλώσσα, η αρχική σελίδα επιλογής επιπέδου ελέγχου καθώς και το «εύκολο» και «μέτριο» επίπεδο δυσκολίας. Παράλληλα, θα πρέπει να γίνει η μετάφραση της τελευταίας σελίδας ελέγχου, αφού όμως πρώτα δημιουργηθεί για τις ανάγκες της παρούσας διπλωματικής εργασίας.



Εικόνα 7: Στιγμιότυπο οθόνης από Notepad++ και σελίδων προγράμματος

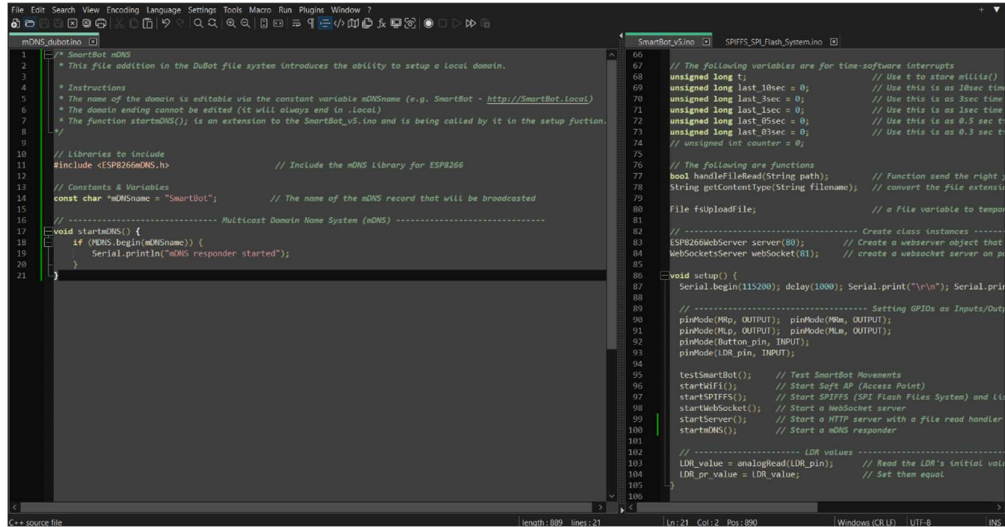
Η ολοκλήρωση της μετάφρασης των ιστοσελίδων από τον συγγραφέα, σε συνεργασία με την προεγκατεστημένη λειτουργία επιλογής γλώσσας από τον Χατζόπουλο κ. συν. (2022) στον πηγαίο κώδικα της ιστοσελίδας, έφερε το πρόγραμμα σε θέση να αξιοποιήσει τις μεταφρασμένες σελίδες χωρίς την ανάγκη αλλαγών στην λειτουργία του προγράμματος.

#### 4.2.2 Προσθήκη Ονομασίας Τοπικού Domain

Ο πηγαίος κώδικας του οχήματος που βασίζεται στην πλακέτα Espressif ESP8266, φέρει δυνατότητες δημιουργίας αυτόνομου τοπικού δικτύου Wi-Fi με την χρήση της ενσωματωμένης κεραίας της πλακέτας ή εξωτερικής κεραίας, όταν τίθεται σε λειτουργία η δυνατότητα Access Point (AP). Σε αντίθετη περίπτωση, η πλακέτα θα έπρεπε να συνδέεται σε οποιοδήποτε διαθέσιμο Wi-Fi μπορεί να υπάρχει στον χώρο, για να μπορεί να ελεγχθεί. Αυτή η επιλογή φέρει την δυνατότητα ελέγχου του οχήματος οπουδήποτε, καθώς δεν υπάρχει ανάγκη εξωτερικού δικτύου και χωρίς επιπλέον περιορισμούς στην ταχύτητα επικοινωνίας.

Η προσθήκη ονομασίας τοπικού domain που προαναφέρθηκε στην ενότητα 3 για τις ανάγκες εύκολης πρόσβασης στην ιστοσελίδα ελέγχου, έναντι στην χρήση της IP διεύθυνσης, μπορεί να επιτευχθεί ανεξάρτητα του τρόπου σύνδεσης της πλακέτας με τον χρήστη, είτε με την προσθήκη εξ ολοκλήρου νέου κομματιού κώδικα στον κώδικα του οχήματος ή με την δημιουργία συνοδευτικού αρχείου το οποίο θα διαχειρίζεται την σύνδεση της πλακέτας και ταυτόχρονα θα καλεί το κατάλληλο πρωτόκολλο επικοινωνίας για την επίτευξη της λειτουργίας αυτής.

Ακολουθώντας τον τρόπο με τον οποίο έχει σχεδιαστή η δομή του πηγαίου κώδικα όπου διαφορετικές λειτουργίες είναι χωρισμένες σε ξεχωριστά αρχεία, ο συγγραφέας επέλεξε να δημιουργήσει ένα ξεχωριστό αρχείο κώδικα που θα αξιοποιεί το multicast DNS πρωτόκολλο, δημιουργώντας ένα mDNS πακέτο με την ονομασία του ιστοτόπου και την συσχέτιση του ονόματος με την IP διεύθυνση της πλακέτας, για τις συσκευές που θα συνδέονται στο ESP8266.



```
mDNS_duobot.ino
1  * SmartBot mDNS
2  * This file addition in the DuBot File system introduces the ability to setup a local domain.
3
4  * Instructions
5  * The name of the domain is editable via the constant variable mDNSName (e.g. SmartBot - http://SmartBot.local)
6  * The domain ending cannot be edited (it will always end in .local)
7  * The function startmDNS(); is an extension to the SmartBot_v5.ino and is being called by it in the setup function.
8
9
10 // Libraries to include
11 #include <ESP8266mDNS.h> // Include the mDNS library for ESP8266
12
13 // Constants & Variables
14 const char mDNSName = "SmartBot"; // The name of the mDNS record that will be broadcasted
15
16 ----- Multicast Domain Name System (mDNS) -----
17 void startmDNS() {
18   if (mDNS.begin(mDNSName)) {
19     Serial.println("mDNS responder started");
20   }
21 }

SmartBot_v5.ino
66 // The following variables are for time-software interrupts
67 unsigned long t; // Use t to store millis()
68 unsigned long last_10sec = 0; // Use this as 10sec time
69 unsigned long last_5sec = 0; // Use this as 5sec time
70 unsigned long last_10sec = 0; // Use this as 10sec time
71 unsigned long last_5sec = 0; // Use this as 0.5 sec t
72 unsigned long last_05sec = 0; // Use this as 0.5 sec t
73 unsigned long last_05sec = 0; // Use this as 0.3 sec t
74 // unsigned int counter = 0;
75
76 // The following are functions
77 bool handleFileRead(String path); // Function send the right
78 String getContentType(String filename); // convert the file extensi
79
80 File fsUploadFile; // a File variable to temp
81
82 ----- Create class instances -----
83 ESP8266Server server(80); // Create a webserver object that
84 WebSocketServer webSocket(81); // create a websocket server on p
85
86 void setup() {
87   Serial.begin(115200); delay(1000); Serial.println("\n\n"); Serial.prin
88
89   // ----- Setting @PIDs as Inputs/Outputs -----
90   pinMode(MP, OUTPUT); pinMode(MB, OUTPUT);
91   pinMode(MLP, OUTPUT); pinMode(MLM, OUTPUT);
92   pinMode(button_pin, INPUT);
93   pinMode(LDR_pin, INPUT);
94
95   testSmartBot(); // Test SmartBot Movements
96   startWiFi(); // Start Soft AP (Access Point)
97   startSPIFFS(); // Start SPIFFS (SPI Flash File System) and its
98   startWebSocket(); // Start a websocket server
99   startServer(); // Start a HTTP server with a file read handler
100  startmDNS(); // Start a mDNS responder
101
102 // ----- I2C values -----
103 I2C_value = analogRead(I2C_pin); // Read the I2C's initial val
104 I2C_pr_value = I2C_value; // Set them equal
105
106 }
```

Εικόνα 8: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης mDNS

Έτσι, όπως και στο παράδειγμα της ενότητας 3.3.2, έγινε η απαραίτητη προσθήκη στον κώδικα, όπου με την χρήση της βιβλιοθήκης *ESP8266mDNS.h* και της εντολής *MDNS.begin()* ο κώδικας που δημιουργήθηκε και παρουσιάζεται μέσω της Εικόνα 8, είναι σε θέση να διαβάσει την IP διεύθυνση που λαμβάνει η πλακέτα ESP8266 είτε από την κάρτα Wi-Fi της, είτε από εξωτερικό Wi-Fi δίκτυο, και στην συνέχεια να αποστείλει συσχέτιση μεταξύ των δύο μέσω του mDNS πακέτου. Το μόνο που χρειάζεται για να λειτουργήσει αυτή η επέκταση, είναι η προσθήκη της λειτουργίας *startmDNS()*; στον πηγαίο κώδικα για να την εκκινήσει κατά την ενεργοποίηση της πλακέτας.

Εδώ είναι χρήσιμο να σημειωθεί πως υπάρχει δυνατότητα με την εντολή *MDNS.begin()* να συνδεθούν και άλλα ονόματα με επιπλέον IP διευθύνσεις στην περίπτωση όπου υπάρχει ανάγκη (λ.χ. Αν το ESP8266 είναι σταθμός και έχει διάφορες IP για διάφορες λειτουργίες – υποσυστήματα), προσθέτοντας εντός της παρενθέσεως το όνομα του Domain και έπειτα την IP στην οποία θα αντιστοιχεί.

### 4.2.3 Προσθήκη ESP32 Πλακέτας & ESP - NOW

Στο επίπεδο της γενικότερης αναβάθμισης των υλικοτεχνικών επιλογών που έχουν γίνει από τους δημιουργούς του ρομποτικού συστήματος, Χατζόπουλος κ. συν. (2022), ερευνήθηκαν κάποιες επιλογές όπως η πλήρη αλλαγή της πλακέτας ESP8266 με πλακέτα περισσότερων δυνατοτήτων, ή την προσθήκη δευτερεύουσας πλακέτας για την ενίσχυση του συστήματος. Η περίπτωση της αλλαγής της πλακέτας ESP8266 με κάποια καλύτερων δυνατοτήτων, είναι πιθανότατα άμεσα εφικτή εάν επιλεγθεί η πλακέτα ESP32. Μια πλακέτα που μπορεί να θεωρηθεί ο διάδοχος της πρώτης, καθώς είναι μια βελτιωμένη και πιο ισχυρή έκδοση του ESP8266, με πρόσθετες δυνατότητες αλλά και συμβατότητα προς την παλαιότερη πλατφόρμα.

Με μερικές αλλαγές στις βιβλιοθήκες που εμπεριέχει ο κώδικας Arduino, στις αντίστοιχες βιβλιοθήκες για το ESP32 και πιθανότατα αλλαγές στις συνδέσεις των εισόδων – εξόδων για να αντιστοιχούν στην μορφολογία της διαφορετικής πλακέτας, το σύστημα θα μπορούσε να προσαρμοστεί στην αλλαγή αυτή. Το κόστος αυτής της αλλαγής είναι αρκετά μικρό, της τάξεως των 10 με 15 ευρώ για μια νέα πλακέτα ESP32, αν και το κόστος έχει άμεση σχέση με την επιλεγμένη υλοποίηση της πλακέτας, διότι υπάρχουν πληθώρα επιλογών ESP32 από ένα πλήθος κατασκευαστών. Η επιλογή αυτή είναι ιδιαίτερα χρήσιμη στην περίπτωση που δεν είναι ήδη αγορασμένη η πλακέτα ESP8266 καθώς και εάν δεν υπάρχει σκοπός για την προσθήκη πολλαπλών αναβαθμίσεων στο σύστημα.

Στην εναλλακτική περίπτωση της προσθήκης δευτερεύουσας πλακέτας στο σύστημα, η επιλογές ξεπερνάνε τα όρια των πλακετών της Espressif (ESP), καθώς στην περίπτωση αυτή δεν υπάρχει ανάγκη αντικατάστασης των λειτουργιών του ESP8266 στα πλαίσια της δευτερεύουσας πλακέτας, παρά μόνο χρειάζεται να μπορεί η δευτερεύουσα πλακέτα να επικοινωνεί με τον ESP8266. Ανάλογα τις επιθυμητές προσθήκες (λ.χ. περισσότεροι ή μεγαλύτεροι κινητήρες για κίνηση, κάμερα, μικρόφωνο κ.α.), την διαθεσιμότητα και το κόστος των πλακετών, οι επιλογές περιορίζονται μόνο από τα πρωτόκολλα που μπορεί να δεχτεί η κεντρική πλακέτα.

Ακολουθώντας την δεύτερη επιλογή με την προσθήκη δευτερεύουσας πλακέτας στο σύστημα του οχήματος, η πλακέτα που επιλέχθηκε από τον συγγραφέα της διπλωματικής εργασίας είναι τύπου ESP32 καθώς είναι θεμιτή η διατήρηση χαμηλού

κόστους αλλά και συνολικού όγκου στο όχημα, ενώ πιθανό είναι σε μελλοντικό στάδιο, μερικές από τις λειτουργίες της κύριας πλακέτας να μεταφερθούν στην δευτερεύουσα για διαμοιρασμό του φόρτου εργασίας. Πιο συγκεκριμένα εξετάστηκαν οι υλοποιήσεις του Πίνακα 2, εκ των οποίων τελικά επιλέχθηκε μία, για την διεξαγωγή δοκιμών και την επιλογή επιπλέον λειτουργιών που θα μπορούσαν να προστεθούν.

Πίνακας 2: Εκλογή ESP32 υλοποίησης ως δευτερεύουσα πλακέτα

Πλακέτα	Πλεονεκτήματα	Κόστος
AI Thinker ESP32 CAM	Κάμερα	12.80€
ESP32-EYE	Κάμερα, Μικρόφωνο	29.00€
ESP32 DevKit V1	Μικρό κόστος	9.90€
Wemos D1 Mini (ESP32 Version)	Μικρός όγκος	19.59€
AI Thinker NodeMCU-32S	Μπορεί να τρέξει MicroPython	11.00€
ESP32-WROOM-32D	USB σύνδεση στην πλακέτα	12.40€

Από τις παραπάνω πλακέτες, ιδιαίτερο ενδιαφέρον φέρουν οι δύο πλακέτες της AI Thinker (ESP32 CAM, NodeMCU-32S), λόγο του χαμηλού κόστους τους σε συνδυασμό με την προσθήκη κάμερας ή την δυνατότητα να λειτουργήσει με κώδικα σε γλώσσα MicroPython αντίστοιχα. Επίσης αρκετό ενδιαφέρον φέρει και η επιλογή του ESP32-EYE που ωστόσο απορρίφθηκε λόγο της υψηλού τιμής του κατά την περίοδο συγγραφής της διπλωματικής. Η τελική επιλογή ανάμεσα στα δύο μοντέλα της AI Thinker έγινε, δίνοντας μεγαλύτερη βάση στην διαθεσιμότητα των δύο πλακετών και των πλεονεκτημάτων τους, όπου η ενσωματωμένη κάμερα της ESP32 CAM, θεωρήθηκε πιο καλή επιλογή.

Ο τρόπος διασύνδεσης των δύο πλακετών θα γίνει με βάση το πρωτόκολλο επικοινωνίας ESP – NOW, ένα πρωτόκολλο επικοινωνίας της κατασκευάστριας εταιρίας Espressif που επιτρέπει την επικοινωνία μεταξύ πλακετών της εταιρίας χωρίς την ανάγκη σύνδεσης σε τοπικό δίκτυο, χρησιμοποιώντας την κεραία της πλακέτας για μεταφορά δεδομένων στην 2.4GHz συχνότητα προς όλες τις πλακέτες που έχουν συζευχθεί με αυτή. Το πλεονέκτημα που προσφέρει αυτή η επιλογή είναι ότι οι δύο πλακέτες θα επωφελούνται άμεσης σύνδεσης χωρίς επιπλέον φυσικά καλώδια, κρατώντας την κατανάλωση ενέργειας στο ελάχιστο.

Από την πλευρά της κεντρικής πλακέτας, η χρήση του πρωτοκόλλου ESP – NOW γίνεται με την προσθήκη της βιβλιοθήκης *espnow.h* και μερικού κώδικα, μέσω του οποίου τυποποιείται η μορφή που θα έχει το πακέτο που θα παραλαμβάνει η πλακέτα ESP8266, καθώς και του κώδικα που θα εκκινεί και θα συσχετίζει τα εισερχόμενα μηνύματα με τον κώδικα.

```

espnow_esp8266_receiver.ino
10
11 // The following is the base data structure, must be the same on the Sender. Used the ba
12
13 #define COMMAND_LENGTH 32
14 char commandName[32];
15 int commandValue;
16 #} CommandData;
17
18 CommandData espData; //Structure object
19
20 // Function to copy all received data to be able to process them
21
22 void onDataReceived(uint8_t * mac, uint8_t *incomingData, uint8_t len) {
23     memcpy(espData, incomingData, sizeof(espData));
24
25     /* To Debug data transfer remove comments on the following lines
26     Serial.print("Bytes received: ");
27     Serial.println(len);
28     Serial.print("Command Name: ");
29     Serial.println(espData.commandName);
30     Serial.print("Command Value: ");
31     Serial.println(espData.commandValue);
32     Serial.println();
33     */
34 }
35
36 // Function to initialize the ESP NOW protocol
37
38 void startESPNow() {
39     if (esp_now_init() != 0) {
40         Serial.println("Encountered error when starting ESP-NOW");
41         return;
42     }
43
44     // Possibly add here code to register Peer if bidirectional communication
45
46     esp_now_register_recv_cb(onDataReceived);
47 }
48
SmartBot_v5.ino
67 // The following variables are for time-software interrupts
68 unsigned long t; // Use t to store millis()
69 unsigned long last_10sec = 0; // Use this to as 10sec time interval
70 unsigned long last_3sec = 0; // Use this to as 3sec time interval
71 unsigned long last_1sec = 0; // Use this to as 1sec time interval
72 unsigned long last_05sec = 0; // Use this to as 0.5 sec time interval
73 unsigned long last_03sec = 0; // Use this to as 0.3 sec time interval
74 // unsigned int counter = 0;
75
76 // The following are functions
77 bool handleFileRead(String path); // Function send the right file to the client (if it
78 String getContentType(String filename); // convert the file extension to the MIME type
79
80 File fsUploadFile; // a file variable to temporarily store the received
81
82 // ----- Create class instances -----
83 ESP8266Server server(80); // Create a webserver object that listens for HTTP request
84 WebSocketServer websocket(81); // create a websocket server on port 81
85
86 void setup() {
87     Serial.begin(115200); delay(1000); Serial.print("\n\n"); Serial.println("SmartBot v4.0");
88
89     // ----- Setting GPIOs as Inputs/Outputs -----
90     pinMode(PIN_P, OUTPUT); pinMode(PIN_A, OUTPUT);
91     pinMode(PIN_M, OUTPUT); pinMode(PIN_L, OUTPUT);
92     pinMode(BUTTON_PIN, INPUT);
93     pinMode(D0_PIN, INPUT);
94
95     testSmartBot(); // Test SmartBot Movements
96     startWiFi(); // Start Soft AP (Access Point)
97     startSPIFFS(); // Start SPIFFS (SPI Flash File System) and list all contents
98     startWebSocket(); // Start a websocket server
99     startServer(); // Start a HTTP server with a file read handler and an upload handler
100    startMdns(); // Start a mdns responder
101    startESPNow(); // Start ESP NOW communication
102
103    // ----- I2C values -----
104    I2C_value = analogRead(I2C_PIN); // Read the I2C's initial value
105    I2C_pr_value = I2C_value; // Set them equal.
106
107
108
length: 6492, lines: 144 Ln:1 Col:1 Pos:1 Unix (L) UTF-8 [NS

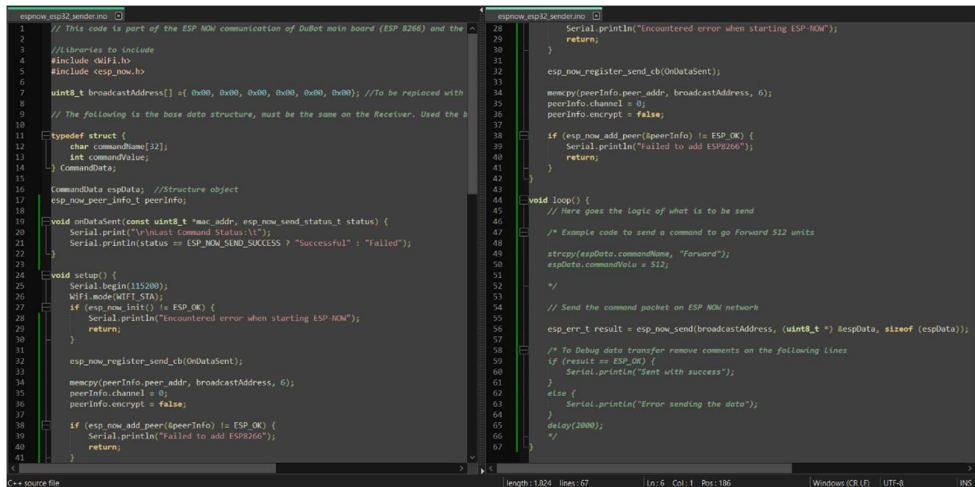
```

Εικόνα 9: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης ESP - NOW

Όπως φαίνεται στην Εικόνα 9, ο κώδικας διαχωρίστηκε σε ξεχωριστό αρχείο όπως και στην περίπτωση του mDNS αν και σε αυτό το σημείο ίσως όλες οι διαδικασίες που σχετίζονται με την επικοινωνία της πλακέτας, θα μπορούσαν να μεταφερθούν σε ένα ενιαίο αρχείο. Στον κώδικα αυτό, το πακέτο δεδομένων που θα λαμβάνει η πλακέτα έχει οριστεί σε δύο κομμάτια με ξεχωριστούς τύπους δεδομένων το καθένα, μια συμβολοσειρά με μέγιστο μέγεθος 32 χαρακτήρων και έναν ακέραιο αριθμό, για να μοιάζει την τυποποιημένη μορφή μιας μονής εντολής (βλ. ενότητα 3.2). Παράλληλα, δημιουργήθηκε η λειτουργία *startESPNow()*; την οποία και καλεί ο κεντρικός κώδικας για να ξεκινήσει το πρωτόκολλο αυτό.

Η επιλογή στην περίπτωση αυτού του κώδικα φέρνει την κεντρική μονάδα ως λήπτη πληροφοριών από την δευτερεύουσα πλακέτα, αν και είναι εφικτό με μικρές προσθήκες, η επικοινωνία να γίνει αμφίδρομη εάν αυτό χρειαστεί από κάποια μελλοντική αναβάθμιση. Για την πιο ολοκληρωμένη εικόνα του κώδικα και της χρήσης του ESP - NOW πρωτοκόλλου, έγινε σχεδιασμός και του αρχείου σε κώδικα Arduino που θα χρειαστεί να τρέχει στην δευτερεύουσα πλακέτα που επιλέχθηκε.

Ο κώδικας αυτός παρουσιάζεται μέσω της Εικόνα 10 και είναι παρόμοιος με αυτόν για την κεντρική πλακέτα. Μικρή διαφορά στην περίπτωση της δευτερεύουσας πλακέτας φέρει ο κώδικας που θα τρέχει σε επανάληψη για να αποστέλλει δεδομένα στην κεντρική πλακέτα, ο οποίος θα πρέπει να διαμορφωθεί ανάλογα με τις όποιες λειτουργίες προστεθούν.



```
1 // This code is part of the ESP-NOW communication of DuBot main board (ESP 8266) and the
2
3 // Libraries to include
4 #include <WiFi.h>
5 #include <esp_now.h>
6
7 uint8_t broadcastAddress[] = { 0x00, 0x00, 0x00, 0x00, 0x00 }; // To be replaced with
8
9 // The following is the base data structure, must be the same on the Receiver. Used the b
10
11 #define struct {
12   char commandName[32];
13   int commandValue;
14   CommandData;
15 }
16 CommandData espData; // Structure object
17 esp_now_peer_info_t peerInfo;
18
19 void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
20   Serial.print("Received Command Status:");
21   Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Successful" : "Failed");
22 }
23
24 void setup() {
25   Serial.begin(115200);
26   WiFi.mode(WIFI_STA);
27   if (esp_now_init() != ESP_OK) {
28     Serial.println("Encountered error when starting ESP-NOW");
29     return;
30   }
31
32   esp_now_register_send_cb(onDataSent);
33
34   memcpy(peerInfo.peer_addr, broadcastAddress, 6);
35   peerInfo.channel = 0;
36   peerInfo.encrypt = false;
37
38   if (esp_now_add_peer(peerInfo) != ESP_OK) {
39     Serial.println("Failed to add ESP8266");
40     return;
41   }
42 }
43
44 void loop() {
45   // Here goes the logic of what is to be send
46
47   /* Example code to send a command to go Forward 512 units
48   */
49   strcpy(espData.commandName, "Forward");
50   espData.commandValue = 512;
51
52   /*
53   */
54   // Send the command packet on ESP-NOW network
55   esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &espData, sizeof(espData));
56
57   /* To Debug data transfer remove comments on the following lines
58   */
59   if (result == ESP_OK) {
60     Serial.println("Sent with success");
61   }
62   else {
63     Serial.println("Error sending the data");
64   }
65   delay(2000);
66 }
67 }
```

Εικόνα 10: Στιγμιότυπο οθόνης από Notepad++ και προσθήκης ESP - NOW σε ESP32

Ως προέκταση του κύριου συστήματος, ο κώδικας αυτός επαρκεί για να φέρει τις δύο πλακέτες σε επικοινωνία, ενώ είναι σε θέση να προστεθούν πλήθος δυνατοτήτων στην δεύτερη πλακέτα ή να διαμοιραστούν υπάρχουσες λειτουργίες, με στόχο την εξισορρόπηση φορτίου μεταξύ των δύο. Από την πλευρά των ηλεκτρονικών, το ESP32 μπορεί να τροφοδοτηθεί με τον ίδιο τρόπο που τροφοδοτείται το ESP8266 εάν εφαρμοστεί τοπικά στο όχημα, ενώ υπό άλλη περίπτωση θα πρέπει να γίνει αντίστοιχα μελέτη και στην συνέχεια σχεδίαση ξεχωριστού κυκλώματος τροφοδοσίας και διασύνδεσης. Κάτι τέτοιο ωστόσο δεν θεωρείται απαραίτητο σε αυτό το σημείο της μελέτης.

### 4.3 2<sup>ος</sup> Κύκλος Βελτιώσεων Συστήματος

Έχοντας καλύψει όλες τις λειτουργίες που αναφέρονται στο κεφάλαιο 3 όπου και έγινε η αποσαφήνιση του συστήματος αναφοράς και η εύρεση των σημείων που χρήζουν προσοχής, πλην της ιστοσελίδας ελέγχου στο «δύσκολο» επίπεδο λειτουργίας, έγινε ένας δεύτερος κύκλος συλλογής ιδεών που αργότερα προστέθηκε στον πίνακα 1 της ενότητας 4.1, προσθέτοντας ιδέες για τις δυνατότητες εξέλιξης



του οχήματος, πάντα με γνώμονα την βελτίωση της εκμάθησης μέσω του συστήματος αναφοράς για μαθητές της πρωτοβάθμιας εκπαίδευσης.

#### **4.3.1 Συνδυασμός Κώδικα Από Edge Impulse**

Μια αρκετά ενδιαφέρουσα προσθήκη στο υπάρχον σύστημα, υπήρξε η συσχέτιση του οχήματος με την πλατφόρμα Edge Impulse και συνεπώς όλες τις λειτουργίες που εμπεριέχει. Η πλατφόρμα αυτή βασίζεται σε πρόγραμμα ανοιχτού κώδικα που μπορεί να εκπαιδεύσει με την χρήση μηχανικής μάθησης, μοντέλα για οπτική ή ηχητική αναγνώριση. Η χρήση της πλατφόρμας είναι δωρεάν προς όλους και ειδικά για μαθητές και φοιτητές, ενώ υπάρχει και δυνατότητα παροχής υπηρεσιών επί πληρωμή. Η διαφορά μεταξύ των δύο είναι το μέγεθος του εκάστοτε μοντέλου που θα εκπαιδευτεί καθώς και οι δυνατότητες που θα φέρει το μηχάνημα – υπολογιστής που θα κάνει την εκπαίδευση.

Χρησιμοποιώντας την πλατφόρμα ως εργαλείο εκμάθησης τόσο για τις μηχανές όσο και για τους μαθητές που θα την χρησιμοποιήσουν, το τελικό μοντέλο που παράγεται μέσω του προγράμματος της πλατφόρμας μετά από παρεμβάσεις του χρήστη στην διαμόρφωση του, μπορεί να επιλεγθεί να εγκατασταθεί σε πηγαίο κώδικα με βάση την πλακέτα στην οποία και θα τρέχει μελλοντικά (λ.χ. βιβλιοθήκη Arduino, βιβλιοθήκη C++, microPython). Η λειτουργία αυτή στην περίπτωση του ρομποτικού συστήματος αναφοράς της μελέτης, μπορεί να προσαρμοστεί με στόχο το μοντέλο οπτικής ή ηχητικής αναγνώρισης να λειτουργεί σε πλακέτες της Espressif ESP32 απευθείας, ενώ παράλληλα η πλατφόρμα είναι σε θέση να υπολογίσει με αρκετή ακρίβεια την τελική κατανάλωση πόρων και τον χρόνο ανά λειτουργία που θα φέρει η πλακέτα που επιλέχθηκε.

Στην περίπτωση πλακετών της Espressif, η διαθέσιμη επιλογή είναι συγκεκριμένα η πλακέτα ESP32-EYE με την επιλογή δημιουργίας βιβλιοθήκης σε γλώσσα Arduino ή σε γλώσσα προγραμματισμού C++. Αν και η πλακέτα που επιλέχθηκε σε προηγούμενη ενότητα (βλ. ενότητα 4.2.3) διαφέρει με την επιλογή αυτή, ο κώδικας που παράγει το Edge Impulse σε Arduino μπορεί να λειτουργήσει και στις δύο, εφόσον επιλεγθεί η κατάλληλη μεταβλητή στην αρχή του κώδικα που παράγει η πλατφόρμα, όπως φαίνεται και στην παρακάτω εικόνα. Στην περίπτωση της πλακέτας που επιλέχθηκε ωστόσο, μόνο προγράμματα που στοχεύουν στην οπτική

αναγνώριση αντικειμένων μπορεί να λειτουργήσουν, καθώς δεν υπάρχει ενσωματωμένο μικρόφωνο, όπως έχει η πλακέτα ESP32-EYE.

```
/* Includes ----- */
#include <SmartBot_inferencing.h>
#include "edge-impulse-sdk/dsp/image/image.hpp"

#include "esp_camera.h"

// Select camera model - find more camera models in camera_pins.h file here
// https://github.com/espressif/arduino-esp32/blob/master/Libraries/ESP32/examples/Camera/CameraWebServer/camera_pins.h
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

Εικόνα 11: Στιγμιότυπο οθόνης από Notepad++ και επιλογής AI Thinker ESP32 -CAM στον κώδικα του Edge Impulse

Στόχος της παρούσας λειτουργίας που αναλύεται σε αυτήν την ενότητα είναι η ευέλικτη προσθήκη του παραγόμενου κώδικα της πλατφόρμας Edge Impulse στο σύστημα αναφοράς, έτσι ώστε οι μαθητές να μπορούν να δημιουργήσουν τα δικά τους μοντέλα οπτικής αναγνώρισης αντικειμένων, τα οποία στην συνέχεια θα μπορούν να εφαρμοστούν στο ρομποτικό όχημα ως μέσο λήψης αποφάσεων και για διάφορους εκπαιδευτικούς σκοπούς. Έτσι, αφού δημιουργήθηκαν μερικές δοκιμαστικές βιβλιοθήκες μέσω της πλατφόρμας Edge Impulse, ο κώδικας τους αναλύθηκε για να βρεθούν τρόποι συνδυασμού του μοντέλου που παράγει η πλατφόρμα με το υπάρχον κώδικα ελέγχου του οχήματος.

Ο κώδικας που δημιουργεί η πλατφόρμα δεν φέρει επιλογές σύνδεσης της πλακέτας με κανένα τρόπο σε άλλη συσκευή, πέραν της σειριακής επικοινωνίας (λ.χ. με χρήση USB καλωδίου στο τερματικό ελέγχου του προγράμματος Arduino IDE), μέσω της οποίας αποστέλλει και τα αποτελέσματα της οπτικής αναγνώρισης. Το κομμάτι του κώδικα που φέρει αυτή την λειτουργία παρουσιάζεται αυτούσιο μέσω της Εικόνα 12 που ακολουθεί.

```
186 // print the predictions
187 ei_printf("Predictions (DSP: %d ms., Classification: %d ms., Anomaly: %d ms.): \n",
188         result.timing.dsp, result.timing.classification, result.timing.anomaly);
189
190 #if EI_CLASSIFIER_OBJECT_DETECTION == 1
191 bool bb_found = result.bounding_boxes[0].value > 0;
192 for (size_t ix = 0; ix < result.bounding_boxes_count; ix++) {
193     auto bb = result.bounding_boxes[ix];
194     if (bb.value == 0) {
195         continue;
196     }
197     ei_printf(" %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\n", bb.label, bb.value, bb.x, bb.y, bb.width, bb.height);
198 }
199 if (!bb_found) {
200     ei_printf(" No objects found!\n");
201 }
202 #else
203 for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
204     ei_printf(" %s: %.5f\n", result.classification[ix].label,
205             result.classification[ix].value);
206 }
207 #endif
```

Εικόνα 12: Στιγμιότυπο οθόνης από Notepad++ και κώδικα του Edge Impulse για εκτύπωση αποτελεσμάτων σε σειριακή θήρα

Όπως φαίνεται μέσω του πηγαίου κώδικα Arduino που δημιούργησε η πλατφόρμα Edge Impulse, γίνεται χρήση της εντολής `ei_printf` που φαίνεται να είναι μια προσαρμοσμένη μορφή της συνάρτησης `printf`, καθώς και πολλαπλών πεδίων από ένα αντικείμενο με όνομα «`bb`», για την παρουσίαση δεδομένων, χρησιμοποιώντας την σειριακή επικοινωνία της πλακέτας. Από τα δεδομένα του αντικειμένου «`bb`», αυτό που φέρει την μεγαλύτερη σημασία στην περίπτωση συνδυασμού του κώδικα με το ρομποτικό σύστημα που εξετάζει η διπλωματική, είναι το «`bb.label`» στο οποίο το μοντέλο οπτικής αναγνώρισης αποθηκεύει το όνομα του αντικειμένου που αναγνωρίζει την κάθε φορά.

Η υπόθεση που εξερευνάται σε αυτό το σημείο είναι εάν μέσω του συνδυασμού της παραπάνω μεταβλητής με τις αποστολές μηνυμάτων μέσω του ESP - NOW πρωτοκόλλου θα μπορούσε η οπτική αναγνώριση να λειτουργήσει σε συνεργασία με το όχημα. Στην περίπτωση όπου τα αντικείμενα φέρουν ονομασίες στο σύστημα αναγνώρισης που αντιστοιχεί σε κάποια κίνηση, ίσως να μπορούσε να χρησιμοποιηθεί αυτούσια η μεταβλητή «`bb.label`» για την επικοινωνία με το όχημα, ενώ στην περίπτωση που διαφέρουν, πιθανότατα θα πρέπει πρώτα να γίνεται επεξεργασία και αντιστοίχιση της ονομασίας του αντικειμένου με μια εντολή κατανοητή προς το ρομποτικό σύστημα.

Για την επίτευξη συνεργασίας του πηγαίου κώδικα του οχήματος με οποιοδήποτε κώδικα δημιουργήσει η πλατφόρμα Edge Impulse, χωρίς να γίνουν ουσιαστικές αλλαγές στην πλευρά του κώδικα της πλατφόρμας, ο κώδικας που δημιουργήθηκε για την πλακέτα ESP32 μεταβάλετε ώστε να μπορεί να αναλάβει σχεδόν εξ ολοκλήρου την διαδικασία συνεργασίας των δύο. Τα σημεία που θα πρέπει να δοθεί βάση κατά τον προγραμματισμό του κώδικα, ορίζονται στην λίστα που ακολουθεί:

- Επικοινωνία μεταβλητής οπτικής αναγνώρισης με αντίστοιχη στο κεντρικό πρόγραμμα του ESP32.
- Δημιουργία αντιστοίχισης η οποία θα πρέπει να μπορεί να αλλαχθεί με απλά βήματα κατά την συγχώνευση από υποψήφιους μελλοντικούς χρήστες, λειτουργώντας ως λεξικό – μεταγλωττιστής για το πρόγραμμα.
- Σχεδίαση της λειτουργίας αποστολής εντολών μέσω ESP – NOW.

Ως πρώτο βήμα σε αυτό το εγχείρημα, σχεδιάζεται ο σκελετός τον οποίο θα φέρει το πρόγραμμα της πλακέτας ESP32, ο οποίος θα πρέπει να είναι διαμορφωμένος με τέτοιο τρόπο ώστε να είναι εύκολη η προσθήκη του κώδικα της πλατφόρμας Edge Impulse.

<i>File_with_ESP32_code</i>	/	<i>espnw_esp32_sender.ino</i>	(Πηγαίος Κώδικας)
		<i>esp32_camera.ino</i>	(Κώδικας από πλατφόρμα)
		<i>src</i>	(Φάκελος για αρχεία πλατφόρμας)

Η διαμόρφωση αυτή φέρει παρόμοια μορφολογία με αυτή του ESP8266, ενώ η ονομασία τόσο του προγράμματος της πλατφόρμας όσο και του φακέλου για τα αρχεία, είναι αντίστοιχη με τα αποτελέσματα που θα φέρει ο παραγόμενος κώδικας της πλατφόρμας Edge Impulse. Αντικαθιστώντας το κενό *esp32\_camera.ino* και μεταφέροντας τα απαραίτητα δεδομένα εντός του φακέλου «*src*» το βάρος συνεργασίας μεταξύ του παραγόμενου κώδικα και του οχήματος πέφτει στον κώδικα που είναι προσαρμοσμένος στο αρχείο *espnw\_esp32\_sender.ino*.

Αναλύοντας περισσότερο το κομμάτι του κώδικα που παρουσιάζεται στην Εικόνα 12 παραπάνω, αξίζει να γίνει αναφορά σε 2 μεταβλητές οι οποίες ορίζονται τοπικά, εντός της επαναλαμβανόμενης λειτουργίας *void loop() {}*, οι οποίες ίσως να μπορούν να χρησιμοποιηθούν για την ολοκλήρωση του παρόντος εγχειρήματος. Κρατώντας τις μεταβολές στο πρόγραμμα της πλατφόρμας Edge Impulse στο ελάχιστο, εάν γίνει ορισμός των μεταβλητών «*bb*» και «*bb\_found*» σε σημείο όπου θα μπορούν να είναι προσβάσιμες από τον πηγαίο κώδικα του *espnw\_esp32\_sender.ino* τότε ίσως να είναι εφικτή η δημιουργία του πρώτου στόχου της παραπάνω λίστας.

Στην μεταβλητή «*bb*» το σημαντικό δεδομένο παραμένει το κελί «*bb.label*» που παρουσιάζει το όνομα του αντικειμένου που αναγνωρίζει το παραγόμενο μοντέλο, ενώ η μεταβλητή «*bb\_found*» είναι σημαντική για την παρακολούθηση των μεταβολών της εξόδου του προγράμματος ως προς το κατά πόσο χρόνο βλέπει κάτι η κάμερα, αλλά και τότε αλλάζει η τιμή του κελιού «*bb.label*». Για να επιτευχθεί αυτό, οι συνολικές μεταβολές του όποιου προγράμματος προκύπτει από την πλατφόρμα Edge Impulse είναι οι ακόλουθες:

1. Αλλαγή της προεπιλεγμένης πλακέτας σε Ai Thinker
2. Ορισμός της μεταβλητής «*bb\_found*» χρησιμοποιώντας «*bool bb\_found = false;*» στην αρχή του προγράμματος.
3. Ορισμός της μεταβλητής «*bb*» χρησιμοποιώντας «*BoundingBox bb;*»

Έτσι, επιλέγεται η σωστή πλακέτα που χρησιμοποιείται στο ρομποτικό σύστημα και ταυτόχρονα επιτυγχάνεται η δυνατότητα προβολής και επεξεργασίας των παραπάνω μεταβλητών στο κύριο πηγαίο κώδικα της πλακέτας ESP32-CAM. Από την πλευρά του κώδικα του ρομποτικού συστήματος θα πρέπει να γίνουν αντίστοιχα μερικές μεταβολές, όπως:

1. Ορισμός της μεταβλητής «*bb\_found*» χρησιμοποιώντας «*extern bool bb\_found;*» ως μεταβλητή ορισμένη από το αρχείο της κάμερας.
2. Ορισμός της μεταβλητής «*bb*» χρησιμοποιώντας «*BoundingBox bb;*».
3. Σχεδιασμός κώδικα που θα χρησιμοποιεί τις μεταβλητές αυτές όπως κριθεί απαραίτητο.

Ακολουθώντας τις υποδείξεις αυτές και χρησιμοποιώντας το αρχείο κώδικα που είχε δημιουργηθεί νωρίτερα για τις ανάγκες αποστολής πακέτων εντολών από την πλακέτα ESP32 προς την πλακέτα ESP8266 με την χρήση του πρωτοκόλλου ESP - NOW, έγινε δημιουργία σχετικού κώδικα ως παράδειγμα μέσω του οποίου καλούνται οι δύο παραπάνω αναφερόμενες μεταβλητές και εν συνεχεία συσχετίζονται μεταξύ τους, με σκοπό την αποστολή του ονόματος του αντικειμένου που βλέπει η κάμερα μέσω της μεταβλητής «*bb.label*» ως command για την κίνηση του οχήματος, όταν η μεταβλητή «*bb\_found*» ισχύει. Οι αλλαγές στον κώδικα παρουσιάζονται μέσω της εικόνας που ακολουθεί, με την προσθήκη σχολιασμού όπου αυτό κρίθηκε απαραίτητο.

```

26 // Constants & Variables
27 const char *ssid = "SmartBot"; // Match the ssid to ESP8266
28 const char *password = "SmartBot"; // Match the password to ESP8266
29 uint8_t broadcastAddress[] = { 0x00, 0x00, 0x00, 0x00, 0x00 }; // To be replaced with MAC Address of ESP8266
30 extern bool bb_found; // External declaration for bb_found
31 BoundingBox bb; // Struct declaration for bb
32 String cam_cmd; // Declare a command variable for the camera label to use in this program
33
72 void loop() {
73
74 // Here goes the logic of what is to be send
75
76 if (bb_found) {
77 cam_cmd = bb.label;
78 } else {
79 cam_cmd = "";
80
81 Serial.println("Command: ");
82 Serial.println(cam_cmd);
83
84 // Prepare the packet
85 strcpy(espData.commandName, cam_cmd.c_str()); // Setting it up to send cam_cmd in the espData packet of ESP NOW
86 espData.commandValue = 1; // 1 is used as an example in this case, holds no importance at this current moment
87
88 // Send the command packet on ESP NOW network
89 esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &espData, sizeof(espData));

```

Εικόνα 13: Στιγμιότυπο οθόνης από Notepad++ και κώδικα παραδείγματος

Με τον παραπάνω κώδικα και τις μικρές αλλαγές στο όποιο αρχείο δημιουργήσει η πλατφόρμα Edge Impulse, είναι δυνατή η οπτική αναγνώριση αντικειμένων και η μεταφορά του ονόματος του αντικειμένου ως εντολή στην πλακέτα ESP8266. Αν τα αντικείμενα που επιλεγθούν φέρουν ονομασίες μέσω του μοντέλου οπτικής αναγνώρισης που αντιστοιχούν σε πραγματικές εντολές του οχήματος (λ.χ. Forward, Backward, Left90 κ.λπ.) και με μικρές αλλαγές όπου αυτό χρειαστεί στο παραπάνω κώδικα (λ.χ. η μεταβλητή τιμή να οριστεί σε μεγαλύτερο αριθμό), το όχημα θα είναι σε θέση να λάβει τον εντοπισμό αυτών των αντικειμένων ως εντολές κίνησης.

Από την πλευρά της πλακέτας ESP8266, θα πρέπει να γίνει διαμόρφωση του πηγαίου κώδικα με στόχο η λαμβανόμενη εντολή να μπορεί να μεταφραστεί ή να χρησιμοποιηθεί αυτούσια ως εντολή κίνησης του οχήματος. Αυτή η λειτουργία είναι αρκετά απλή στην διαμόρφωση της και μπορεί να επιτευχθεί χρησιμοποιώντας την εντολή *if*, συσχετίζοντας την συμβολοσειρά που φέρει κάθε φορά το μήνυμα του πρωτοκόλλου ESP – NOW με εντολές κίνησης.

(A)

```

22 void onDataReceived(uint8_t * mac, uint8_t *incomingData, uint8_t len) {
23     memcpy(&espData, incomingData, sizeof(espData));
24
25     if (espData.commandName = "Command #1") { /* Code here to run when "Command #1" is send */ }
26     if (espData.commandName = "Command #2") { /* Code here to run when "Command #2" is send */ }
27     if (espData.commandName = "Command #3") { /* Code here to run when "Command #3" is send */ }
28     if (espData.commandName = "Command #4") { /* Code here to run when "Command #4" is send */ }
29
30     /* To Debug data transfer remove comments on the following lines
31     Serial.println("Bytes recieved: ");
32     Serial.println(len);
33     Serial.println("Command Name: ");
34     Serial.println(espData.commandName);
35     Serial.println("Command Value: ");
36     Serial.println(espData.commandValue);
37     Serial.println();
38     */
39 }

```

(B)

```

10 void Movements(int robot_mov, int newspeed) { // Drive robot with appropriate movement & speed
11     if (robot_mov == 0) { Stop(); }
12     if (robot_mov == 1) { Forward(newspeed); }
13     if (robot_mov == 2) { Backward(newspeed); }
14     if (robot_mov == 3) { RotateCW(newspeed); }
15     if (robot_mov == 4) { RotateCCW(newspeed); }
16     if (robot_mov == 5) { FwdRight(newspeed); }
17     if (robot_mov == 6) { FwdLeft(newspeed); }
18     if (robot_mov == 7) { BwdRight(newspeed); }
19     if (robot_mov == 8) { BwdLeft(newspeed); }
20     if (robot_mov == 9) { Left(newspeed); }
21     if (robot_mov == 10) { Right(newspeed); }
22
23
24 // ----- SmartBot Moving Functions with speed control -----
25 void Stop() {
26     Serial.println("Stop Moving");
27     digitalWrite (MRp, LOW); digitalWrite (MRm, LOW); digitalWrite (MLp, LOW); digitalWrite (MLm, LOW);
28     robotMoveNum = 0; }
29
30 void Forward(int sp) {
31     Serial.print("Moving Forward with Speed: "); Serial.println(sp);
32     digitalWrite (MRp, sp); digitalWrite (MRm, LOW); digitalWrite (MLp, LOW); digitalWrite (MLm, sp);
33     robotMoveNum = 1; }

```

Εικόνα 14: Στιγμιότυπο οθόνης από Notepad++ και κώδικα ESP8266 για συσχέτιση εντολών

Η Εικόνα 14 (Α) παρουσιάζει τον σχετικό αυτό κώδικα, όπου η εντολή *if* ελέγχει το περιεχόμενο της μεταβλητής «*espData.commandName*» που μεταφέρει το ESP – NOW πρωτόκολλο και επιτρέπει ξεχωριστό κώδικα να τρέχει για κάθε εντολή που φέρει τιμή αντίστοιχη σε αυτές που ελέγχει η *if*. Ο κώδικας εντός των αγκυλών που έχει αφαιρεθεί σκόπιμα ως σχόλιο μπορεί μετέπειτα να σχεδιαστεί ανάλογα των λειτουργιών που θα προστεθούν από τους χρήστες του οχήματος με βάση τις ανάγκες τους.

Παράλληλα, στην περίπτωση όπου οι εντολές που θα αποστέλλονται μέσω του ESP – NOW δικτύου αφορούν την κίνηση του οχήματος, τότε προτείνεται η σύνδεση της παραπάνω ελεγκτικής λειτουργίας και του προγράμματος που θα προστεθεί για κάθε περίπτωση, με τις εντολές που εμπεριέχει το αρχείο *Movement\_Functions.ino*, όπως αυτές παρουσιάζονται μέσω της Εικόνα 14 (Β).

#### **4.3.2 Επεξεργαστής Γραπτού Κώδικα**

Με την μόνη ανοιχτή εκκρεμότητα ως προς τις απαραίτητες βελτιώσεις του συστήματος αναφοράς να είναι η ολοκλήρωση της ιστοσελίδας ελέγχου και μάλιστα στο «δύσκολο» επίπεδο, το μεγαλύτερο μέρος των ιδεών του δεύτερου κύκλου αφιερώθηκε στην σχεδίαση της σελίδας. Αποσκοπώντας στην διαφοροποίηση της σε σχέση με τις προηγούμενες δύο, μια από τις ιδέες για τον τρόπο διαμόρφωσης της ιστοσελίδας είναι αυτή της προσθήκης επεξεργαστή γραπτού κώδικα (Code Editor), μέσω του οποίου οι μαθητές θα μπορούν να γράφουν πηγαίο κώδικα για τον έλεγχο κινήσεων του οχήματος. Για την επίτευξη προσθήκης του επεξεργαστή στην ιστοσελίδα, έγινε επιλογή χρήσης της βιβλιοθήκης CodeMirror, που μεταξύ άλλων επιτρέπει την δημιουργία επεκτάσεων σε ιστοσελίδες που ομοιάζουν επεξεργαστές γραπτού κώδικα.

Η βιβλιοθήκη CodeMirror είναι βιβλιοθήκη σε γλώσσα JavaScript που επιτρέπει την ενσωμάτωση των επεξεργαστών πηγαίου κώδικα σε διαδικτυακές εφαρμογές, έχει την δυνατότητα να υποστηρίξει πλήθος γλωσσών προγραμματισμού καθώς επίσης ψευδογλώσσες αλλά και προσαρμοσμένες γλώσσες, εφόσον πληρούνται οι όποιες απαιτήσεις του εκάστοτε προγράμματος. Το εγχείρημα αυτό αποδείχθηκε αρκετά απαιτητικό στην δημιουργία του σε συνδυασμό με το ήδη υπάρχον σύστημα του ESP8266, ωστόσο μετά από αρκετές δοκιμές πηγαίου κώδικα και την λήψη των

κατάλληλων αρχείων για την χρήση της βιβλιοθήκης χωρίς την ανάγκη σύνδεσης στο διαδίκτυο, οι πρώτοι δοκιμαστικοί κώδικες που σχεδιάστηκαν φέρανε αρκετά υποσχόμενα αποτελέσματα.

Όπως φαίνεται και στην Εικόνα 15, η δημιουργία ενός επεξεργαστή κώδικα εντός ιστοσελίδας πέραν από την χρήση της βιβλιοθήκης CodeMirror χρειάζεται και μια περιοχή για να εισάγει ο χρήστης κείμενο. Χρησιμοποιώντας την εντολή `<textarea>...</textarea>` της γλώσσας HTML, ο χρήστης μπορεί να γράφει το κείμενο του ως έχει, με την βιβλιοθήκη έπειτα να το μορφοποιεί με τέτοιο τρόπο που οπτικά ομοιάζει με επεξεργαστή κώδικα.



The image shows a screenshot of a web browser window. At the top, there is a title bar that says "Editor Window". Below it is a text area containing the following code:

```
1 testing code
2 speed(500) #speed from 1 to 1023
3 for (i=0; i<4; i++) {
4   forward(100)
5   right(90)
6 }
7
```

Below the code editor is a larger rectangular area that displays the rendered output of the code. The output is:

```
testing code
speed(500) #speed from 1 to 1023
for (i=0; i<4; i++) {
  forward(100)
  right(90)
}
```

At the bottom left of the rendered area, there is a small button labeled "Run".

Εικόνα 15: Στιγμιότυπο οθόνης από CodeMirror σε δοκιμαστική ιστοσελίδα

Στο παράδειγμα της δοκιμαστικής ιστοσελίδας στην Εικόνα 15, έχει σχεδιαστεί η μορφοποίηση του παραθύρου του επεξεργαστή γραπτού κώδικα χωρίς να εφαρμοστεί επισήμανση κάποιας γλώσσας προγραμματισμού. Παράλληλα το αποτέλεσμα του επεξεργαστή κώδικα παρουσιάζεται στο κάτω μέρος της ιστοσελίδας σε δικό του παράθυρο όπου θα μπορούσε να έχει μεταγλωττιστεί σε εντολές για την κίνηση του οχήματος αναφοράς. Σε μια ολοκληρωμένη ιστοσελίδα, η έξοδος του αποτελέσματος δεν θα είναι εμφανής κατά αυτό τον τρόπο, με τους χρήστες να εισάγουν τις εντολές τους μέσω του επεξεργαστή πηγαίου κώδικα και τα αποτελέσματα του κώδικα να γίνονται εμφανή μέσω άλλων λειτουργιών της σελίδας.



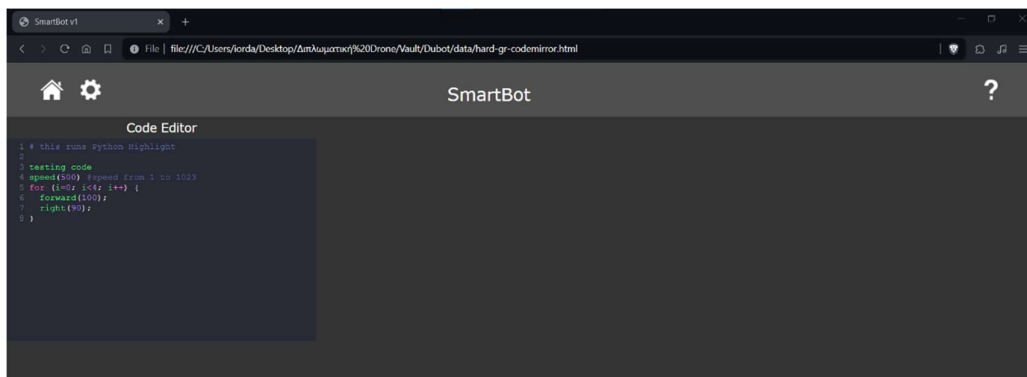
Εφόσον γίνει χρήση του CSS αρχείου που ακολουθεί την βιβλιοθήκη του CodeMirror για τον χρωματισμό των εντολών βάσει μιας γλώσσας προγραμματισμού (λ.χ. C++, Python κ.α.), ο κώδικας επισημαίνεται με χρωματισμούς όπως φαίνεται στην Εικόνα 16.

Editor Window

```
1 testing code
2 speed(500) #speed from 1 to 1023
3 for (i=0; i<4; i++) {
4   forward(100)
5   right(90)
6 }
7
```

Εικόνα 16: Στιγμιότυπο οθόνης από CodeMirror με CSS σε δοκιμαστική ιστοσελίδα

Τόσο τα χρώματα των αναγνωρισμένων εντολών όσο και ποιες εντολές θα αναγνωριστούν είναι αποτέλεσμα των επιμέρους βιβλιοθηκών που έχουν χρησιμοποιηθεί, και που θα μπορούσαν να παραμετροποιηθούν ανάλογα με τις ανάγκες του τελικού συστήματος μέσω του πηγαίου κώδικα. Το παραπάνω παράδειγμα μετά από μερική επεξεργασία εφαρμόστηκε στην ιστοσελίδα που θα φιλοξενεί το «δύσκολο» επίπεδο ελέγχου και παρουσιάζεται στην Εικόνα 17.



Εικόνα 17: Στιγμιότυπο οθόνης από CodeMirror με CSS Python σε «Δύσκολο» επίπεδο ελέγχου

Η προσθήκη του επεξεργαστή κειμένου με βάση το CodeMirror στην ιστοσελίδα αποτελεί την αρχή αυτής της ιστοσελίδας, ωστόσο δεν επαρκεί από μόνο του για την χρήση του στον έλεγχο του οχήματος. Πέραν του πηγαίου κώδικα που χρειάστηκε για την δημιουργία της ιστοσελίδας που παρουσιάζεται στην Εικόνα 17, θα πρέπει

να γίνει σχεδίαση ενός επιπλέον αρχείου κώδικα σε γλώσσα προγραμματισμού JavaScript, περιέχοντας ένα πλήθος λειτουργιών που θα τρέχουν στο παρασκήνιο.

Οι κύριες λειτουργίες που θα πρέπει να προγραμματιστούν στον κώδικα που θα τρέχει στο παρασκήνιο είναι αυτές της λογικής ανάλυσης των όσων γράφονται στον επεξεργαστή κώδικα και η μετέπειτα λογική εκτέλεση του προγράμματος. Η λογική ανάλυση θα είναι υπεύθυνη για την ανάλυση του κώδικα που είναι γραμμένος στον επεξεργαστή CodeMirror και την εξαγωγή των σχετικών πληροφοριών (λ.χ. ορισμοί συναρτήσεων, κλήσεις εντολών, παραμετροποίηση μεταβλητών τιμών κ.α). Ενώ, η λογική εκτέλεση θα είναι υπεύθυνη για την πραγματική εκτέλεση των ενεργειών που καθορίζονται στον αναλυόμενο κώδικα, μετατρέποντας τις όσες εντολές χρησιμοποιούνται σε εντολές που αντιλαμβάνεται ήδη η πλακέτα ESP8266.

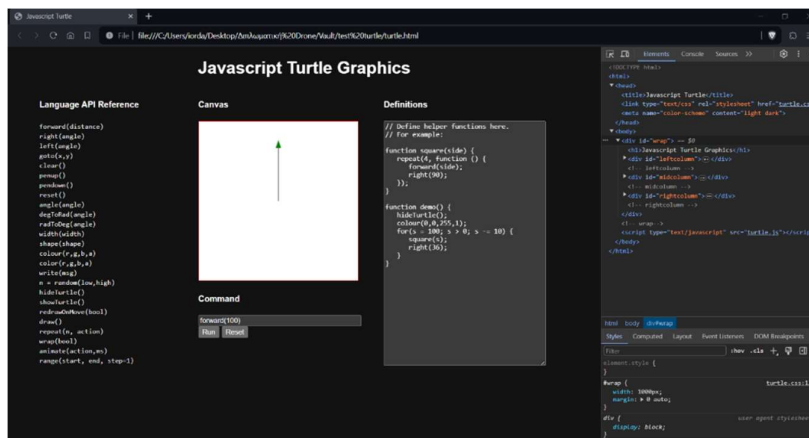
Ο κώδικας που θα υποστηρίζει τις παραπάνω εντολές θα πρέπει να σχεδιαστεί με τέτοιο τρόπο ώστε να είναι σε θέση να λειτουργήσει τόσο για τις ανάγκες του CodeMirror επεξεργαστή, όσο και για την μετατροπή και εκτέλεση των εντολών με τέτοιο τρόπο ώστε να είναι κατανοητό στο όχημα αναφοράς. Παράλληλα, όσες άλλες προσθήκες προκύπτουν στην σελίδα αυτή θα πρέπει επίσης να συνυπολογίζονται σε αυτό. Για τους παραπάνω λόγους ο κώδικας αυτός θα αναλυθεί περισσότερο σε ξεχωριστή ενότητα που ακολουθεί, μετά την προσθήκη των υπόλοιπων επιμέρους λειτουργιών για το «δύσκολο» επίπεδο ελέγχου.

### **4.3.3 Προσθήκη Turtle Graphics**

Στις σημειώσεις που βρίσκονται διαθέσιμες εντός του πηγαίου κώδικα για το ρομποτικό σύστημα αναφοράς, υπάρχουν αναφορές σχετικά με την επιθυμία προσθήκης ενός τρόπου προβολής της κίνησης του οχήματος προτού αυτή πραγματοποιηθεί, συνδυαστικά με λειτουργίες που θα επιτρέπουν την σχεδίαση της κίνησης μέσω πολλαπλών εντολών. Ασχέτως του τρόπου ελέγχου του οχήματος, η λειτουργία που θα πρέπει να δημιουργηθεί θα επιτρέπει κατά μια έννοια την σχεδίαση του πλάνου κίνησης και την προβολή αυτού προτού το όχημα προχωρήσει στην εκτέλεση των κινήσεων που θα εμπεριέχει.

Χρησιμοποιώντας τον επεξεργαστή γραπτού κώδικα CodeMirror που αναλύθηκε σε προηγούμενη ενότητα για τον σχεδιασμό του πλάνου κίνησης μέσω των εντολών

ελέγχου της πλακέτας, έγινε έρευνα για τον τρόπο με τον οποίο θα μπορούσε να γίνει παραμετροποίηση του προβαλλόμενου αποτελέσματος εντός της ιστοσελίδας με τέτοιο τρόπο ώστε να καλύπτει αυτήν την ανάγκη. Για την παρουσίαση των κινήσεων και ουσιαστικά την προσομοίωση του πλάνου κίνησης που θα φέρει το όχημα, έγινε επιλογή της επέκτασης Turtle Graphics, μιας βιβλιοθήκης που αποτελείται από συνδυασμό εντολών κίνησης και γλώσσας προγραμματισμού Python για την σχεδίαση γραφικών.

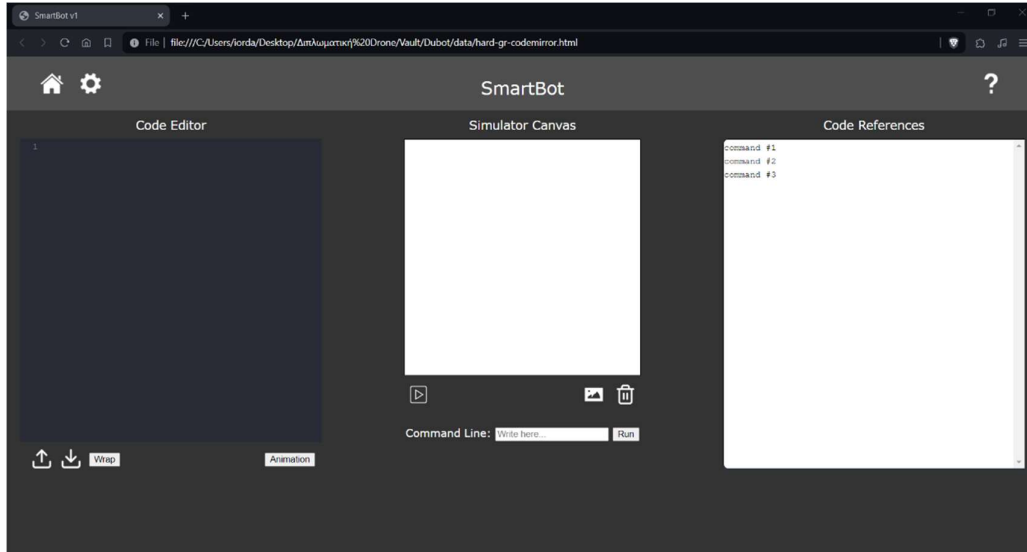


Εικόνα 18: Στιγμιότυπο οθόνης από ιστοσελίδα με JavaScript και Turtle Graphics

Πιο συγκεκριμένα για να προστεθεί η λειτουργία αυτή στην ιστοσελίδα του ρομποτικού συστήματος και ειδικά χωρίς την ανάγκη σύνδεσης στο διαδίκτυο, έγινε έρευνα και λήψη μιας αντίστοιχης βιβλιοθήκης Python Turtle Graphics σε γλώσσα JavaScript που μπορεί να τρέχει στο παρασκήνιο της ιστοσελίδας, καθώς στην περίπτωση του προγράμματος του ESP8266, η χρήση καθαντού της Python δεν είναι εφικτή. Στην Εικόνα 18 παρουσιάζεται το παράδειγμα που υπάρχει διαθέσιμο μέσω του GitHub σχετικά με την χρήση Turtle Graphics, όπου μέσω του κώδικα HTML, των στυλιστικών επιλογών σε CSS και του πηγαίου κώδικα turtle.js, η ιστοσελίδα είναι σε θέση να σχεδιάζει σε ένα καμβά τις κινήσεις της χελώνας, όπως αυτές δίνονται μέσω της γραμμής εντολών.

Ακολούθησαν δοκιμές ως προς την χρήση της βιβλιοθήκης αυτής και ειδικά σε συνδυασμό με τον επεξεργαστή κώδικα που προαναφέρθηκε στην προηγούμενη ενότητα, δημιουργώντας την τελική μορφή που θα φέρει το «δύσκολο» επίπεδο ελέγχου του οχήματος (βλ. Εικόνα 19). Αν και οπτικά η σελίδα είναι έτοιμη για να

χρησιμοποιηθεί, απομένει να γίνει σχεδίαση και προγραμματισμός όλων των επί μέρους λειτουργιών που θα εφαρμόζονται στο παρασκήνιο για να είναι σε θέση η ιστοσελίδα να χειρίζεται το όχημα κατά τον επιθυμητό τρόπο.



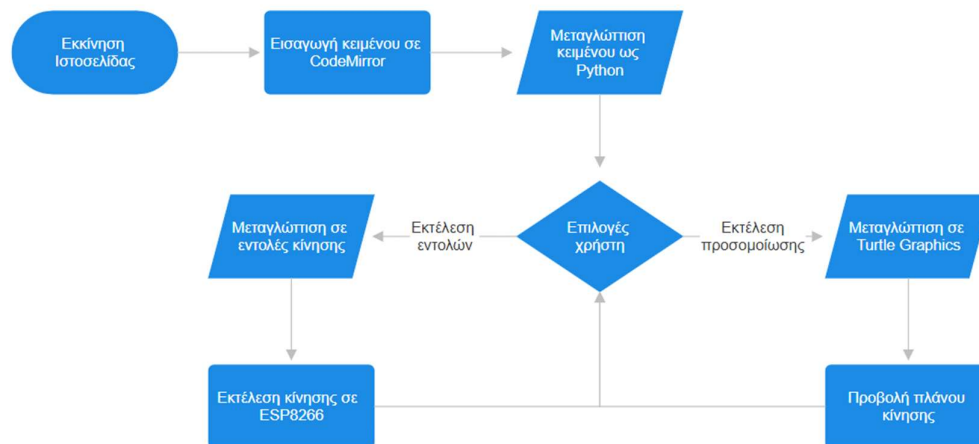
Εικόνα 19: Στιγμιότυπο οθόνης από ιστοσελίδα οχήματος με Turtle Graphics

Η βιβλιοθήκη CodeMirror μπορεί να προσαρμοστεί με τέτοιο τρόπο ώστε να διαβάσει τις εντολές της Python για τα Turtle Graphics μέσω της ιστοσελίδας και με την χρήση της βιβλιοθήκης Turtle σε JavaScript αυτά να εμφανίζονται στον καμβά. Παράλληλα θα πρέπει να γίνει πρόβλεψη για τις τυπικές εντολές επανάληψης και ελέγχου, καθώς και προσθήκη των ειδικών εντολών κίνησης του οχήματος στην περίπτωση όπου ο χρήστης επιλέξει την εκτέλεση του πλάνου κίνησης.

#### 4.3.4 Βιβλιοθήκη Skulpt σε JavaScript

Αναφορικά με τον τρόπο με τον οποίο θα γίνεται η προσαρμογή του κώδικα που γράφει ο χρήστης στον επεξεργαστή CodeMirror σε εντολές κατανοητές τόσο για το όχημα όσο και για την χελώνα της Python για την δημιουργία του γραφικού κομματιού, επιλέχθηκε η προσθήκη ενός διάυλου μεταγλώττισης και επικοινωνίας μεταξύ των παραπάνω προγραμμάτων. Στόχος αυτής της προσθήκης είναι η επίτευξη προσθήκης προχωρημένων λογικών εντολών και λειτουργιών, όμοιων με αυτές των γνωστών γλωσσών προγραμματισμού, ενώ παράλληλα θα είναι εφικτή η χρήση του επεξεργαστή τόσο για το όχημα όσο και για την προσομοίωση, χωρίς αυτές οι δύο λειτουργίες να εξαρτώνται άμεσα η μία από την άλλη.

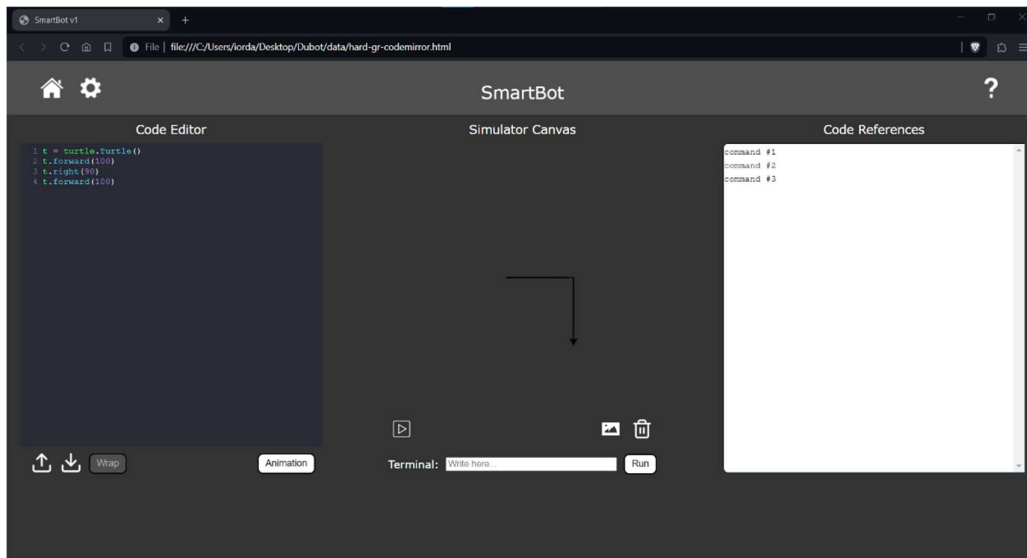
Αν και αρχικά επιλέχθηκε η χρήση της βιβλιοθήκης *turtle.js* για την απλή λήψη των εντολών από τον επεξεργαστή CodeMirror και την προβολή του αποτελέσματος μέσω κινήσεων χελώνας, η επιλογή αυτή δεν παρέχει όλο το πλήθος των εντολών που εμπεριέχει η ολοκληρωμένη γλώσσα προγραμματισμού Python, παρά μόνο τις απαραίτητες εντολές για την παρουσίαση του αποτελέσματος. Για την αύξηση των δυνατοτήτων του επεξεργαστή, έγινε επιλογή της βιβλιοθήκης Skulpt που παρουσιάζεται ως μια επιλογή στον πίνακα 1. Είναι κατασκευασμένη σε γλώσσα προγραμματισμού JavaScript και μεταξύ άλλων επιτρέπει την χρήση γλώσσας προγραμματισμού Python κατευθείαν στην ιστοσελίδα ελέγχου του οχήματος, εμπεριέχοντας παράλληλα την βιβλιοθήκη *turtle.js* εντός αυτής. Η επιλογή αυτή συνδυαστικά με την χρήση γραφικών χελώνας ως προεπιλογή, θα μπορούσε να δημιουργήσει ένα ενδιαφέρον περιβάλλον ελέγχου του οχήματος, εισάγοντας τους μαθητές στον προγραμματισμό με την χρήση γραπτών εντολών.



Εικόνα 20: Λογικό διάγραμμα λειτουργίας σελίδας για «δύσκολο» επίπεδο

Η επιλογή της βιβλιοθήκης Skulpt παρουσιάζεται μέσω του λογικού διαγράμματος στην Εικόνα 20, εξηγώντας τον τρόπο με τον οποίο θα λειτουργεί η ιστοσελίδα με όλες τις προσθήκες που έχουν επιλεγεί να δημιουργηθούν. Ο κώδικας της βιβλιοθήκης Skulpt θα φέρει μεταξύ άλλων την λειτουργία της λογικής ανάλυσης αλλά και της λογικής εκτέλεσης που χρειάζεται το πρόγραμμα, ενώ ταυτόχρονα θα εισάγει όλες τις βασικές λειτουργίες της γλώσσας Python, όπως εντολές επανάληψης ή λήψης αποφάσεων κ.α.. Σε συνεργασία με την επιλογή αυτή έγινε προσθήκη και της βιβλιοθήκης jQuery που εμπλουτίζει τον κώδικα της γλώσσας JavaScript, με

διάφορες δυνατότητες, για τον καλύτερο έλεγχο του HTML ιστοτόπου και την σχεδίαση των υποσυστημάτων της σελίδας.



Εικόνα 21: Στιγμιότυπο οθόνης από ιστοσελίδα οχήματος με Turtle Graphics & Skulpt

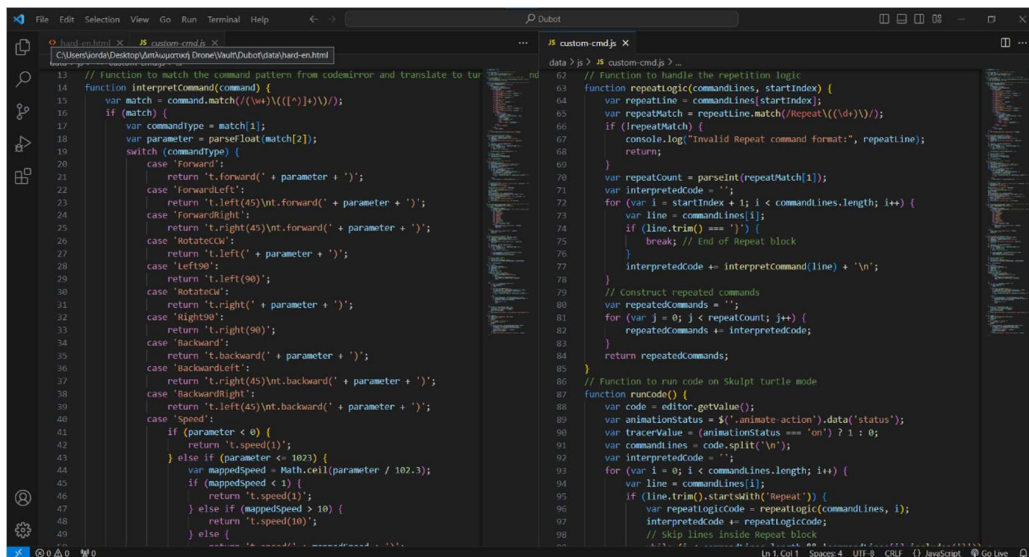
Στην Εικόνα 21 παρουσιάζεται η ιστοσελίδα του ρομποτικού συστήματος που έχει σχεδιαστεί από τον συγγραφέα της διπλωματικής εργασίας, χρησιμοποιώντας όλες τις παραπάνω αναφερόμενες βιβλιοθήκες. Ο CodeMirror επεξεργαστής λαμβάνει τις εντολές από τον χρήστη ως απλό κείμενο, το οποίο μορφοποιείται τόσο σε γλώσσα προγραμματισμού Python όσο και συγκεκριμένα για την κίνηση της χελώνας μέσω της ολοκληρωμένης βιβλιοθήκης Skulpt και στην συνέχεια παρουσιάζει τα αποτελέσματα της εκτέλεσης του κώδικα, στον προκαθορισμένο χώρο στο κέντρο της σελίδας που ορίστηκε ως καμβάς για την προεπισκόπηση των κινήσεων.

Σε αυτό το σημείο απομένει η προσθήκη της λειτουργίας μεταγλώττισης των εντολών χελώνας σε εντολές που να αντιλαμβάνεται το όχημα αναφοράς, καθώς επίσης η προσθήκη των λειτουργιών για το σύνολο των πλήκτρων της ιστοσελίδας και του τερματικού για την αποστολή εντολών. Τόσο ο πηγαίος κώδικας που θα μεταγλωττίζει τις εντολές του χρήστη σε εντολές για το όχημα όσο και η λογική που θα μετατρέπει το σύνολο των εντολών σε πακέτο εντολοσειράς (βλ. Εικόνα 3), αναλύονται στην ενότητα 4.3.5 που ακολουθεί, ολοκληρώνοντας παράλληλα το σύνολο των βελτιώσεων και αναβαθμίσεων του συστήματος που επιλέχθηκε στην αρχή της διπλωματικής εργασίας.

### 4.3.5 Ολοκλήρωση Ιστοσελίδας

Έχοντας ολοκληρώσει τις βασικές λειτουργίες που θα εμπεριέχει η ιστοσελίδα στις υποενότητες της ενότητας 4.3, σειρά έχει ο τελικός συνδυασμός τους σε ένα αρχείο καθώς επίσης η προσθήκη συνοδευτικού κώδικα για τα πλήκτρα και τις λοιπές λειτουργίες που θα πρέπει να τρέχουν στο παρασκήνιο του ιστοτόπου. Κατά την τελική συναρμολόγηση των λειτουργιών και την προσθήκη της λογικής πίσω από τον τρόπο επικοινωνίας με την πλακέτα ελέγχου του οχήματος, παρατηρήθηκαν μερικά προβλήματα στον τρόπο λειτουργίας των βιβλιοθηκών που έχουν επιλεγεί που πιθανότατα να οφείλονται σε ασυμβατότητες μεταξύ των βιβλιοθηκών.

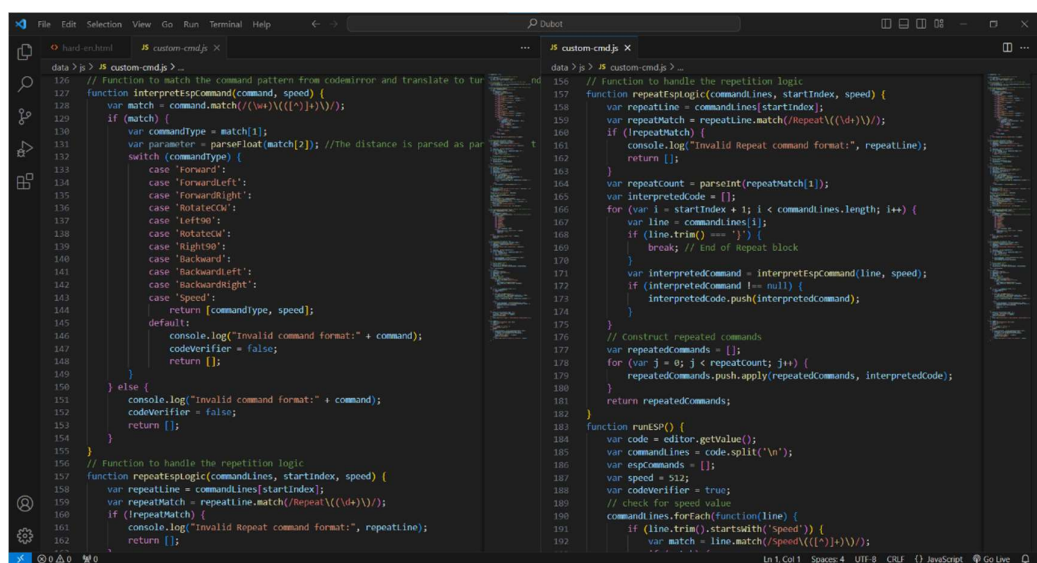
Κάποια από τα προβλήματα φαίνεται να πηγάζουν στην σειρά με την οποία επεξεργάζεται η ιστοσελίδα και κατά συνέπεια οι βιβλιοθήκες τα δεδομένα που εισάγει ο τελικός χρήστης μέσω του CodeMirror επεξεργαστή. Για την λύση του προβλήματος αυτού επιλέχθηκε τελικά να αποσπαστεί η απευθείας σύνδεση του CodeMirror επεξεργαστή με την βιβλιοθήκη Skulpt, εισάγοντας ως ενδιάμεσο βήμα την λήψη του κειμένου και την αποθήκευση του σε μια μεταβλητή αναπαράσταση μιας λίστας από αλφαριθμητικές ακολουθίες χαρακτήρων. Με την επιλογή αυτή, ο πηγαίος κώδικας είναι σε θέση να επεξεργαστεί το σύνολο των εντολών του χρήστη γραμμή προς γραμμή, αντί του συνόλου του κειμένου.



```
13 // function to match the command pattern from codemirror and translate to turtle
14 function interpretCommand(command) {
15   var match = command.match(/((w+)(([0-9]+)?)\s?)/);
16   if (match) {
17     var commandType = match[1];
18     var parameter = parseInt(match[2]);
19     switch (commandType) {
20       case 'forward':
21         return 't.forward(' + parameter + ')';
22       case 'forwardleft':
23         return 't.left(45).forward(' + parameter + ')';
24       case 'forwardright':
25         return 't.right(45).forward(' + parameter + ')';
26       case 'rotatereel':
27         return 't.left(' + parameter + ')';
28       case 'left90':
29         return 't.left(90)';
30       case 'rotatetow':
31         return 't.right(' + parameter + ')';
32       case 'right90':
33         return 't.right(90)';
34       case 'backward':
35         return 't.backward(' + parameter + ')';
36       case 'backwardleft':
37         return 't.right(45).backward(' + parameter + ')';
38       case 'backwardright':
39         return 't.left(45).backward(' + parameter + ')';
40       case 'speed':
41         if (parameter < 0) {
42           return 't.speed(1)';
43         } else if (parameter <= 1023) {
44           var mappedSpeed = Math.ceil(parameter / 102.3);
45           if (mappedSpeed < 3) {
46             return 't.speed(3)';
47           } else if (mappedSpeed > 10) {
48             return 't.speed(10)';
49           } else {
50             return 't.speed(10)';
51           }
52         }
53     }
54   }
55 }
56
57 // Function to handle the repetition logic
58 function repeatLogic(commands, startIndex) {
59   var repeatLine = commands[startIndex];
60   var repeatMatch = repeatLine.match(/repeat((\d+)\s?)/);
61   if (!repeatMatch) {
62     console.log("Invalid Repeat command format:", repeatLine);
63     return;
64   }
65   var repeatCount = parseInt(repeatMatch[1]);
66   var interpretedCode = '';
67   for (var i = startIndex + 1; i <= commands.length; i++) {
68     var line = commands[i];
69     if (line.trim() === '') {
70       break; // End of Repeat block
71     }
72     interpretedCode += interpretCommand(line) + '\n';
73   }
74   // Construct repeated commands
75   var repeatedCommands = '';
76   for (var j = 0; j < repeatCount; j++) {
77     repeatedCommands += interpretedCode;
78   }
79   return repeatedCommands;
80 }
81
82 // Function to run code on Skulpt turtle mode
83 function runCode() {
84   var code = editor.getValue();
85   var animationStatus = $.animate.action().data('status');
86   var traceValue = (animationStatus === 'on') ? 1 : 0;
87   var commandLines = code.split('\n');
88   var interpretedCode = '';
89   for (var i = 0; i <= commandLines.length; i++) {
90     var line = commandLines[i];
91     if (line.trim().startsWith('repeat')) {
92       var repeatLogicCode = repeatLogic(commandLines, i);
93       interpretedCode += repeatLogicCode;
94     } else {
95       // skip lines inside repeat block
96     }
97   }
98 }
```

Εικόνα 22: Στιγμιότυπο οθόνης από κώδικα ιστοσελίδας με μεταγλωττιστη εντολών χρήστη σε εντολές χελώνας

Παράλληλα, έγινε εισαγωγή προσαρμοσμένων εντολών (custom commands) στην λογική του CodeMirror επεξεργαστή έτσι ώστε ο χρήστης να μπορεί να προγραμματίζει το πλάνο κίνησης με τις ίδιες εντολές που θα χρησιμοποιούσε στα προηγούμενα επίπεδα ελέγχου. Τόσο οι προσαρμοσμένες εντολές όσο και η λογική ελέγχου και μεταγλώττισης αυτών, μεταφέρθηκαν σε ξεχωριστό από την ιστοσελίδα αρχείο σε γλώσσα προγραμματισμού JavaScript το οποίο καλείται από την ιστοσελίδα κατά την εκκίνηση της, και παρουσιάζεται στην Εικόνα 22. Στο ίδιο αρχείο προστέθηκε παράλληλα και η μεταγλώττιση των εντολών σε πακέτα που να αντιλαμβάνεται ο διακομιστής της ιστοσελίδας για την αποστολή των κινήσεων στην πλακέτα ESP8266. Ο σχετικός κώδικας με την λειτουργία αυτή παρουσιάζεται μέσω της Εικόνα 23.



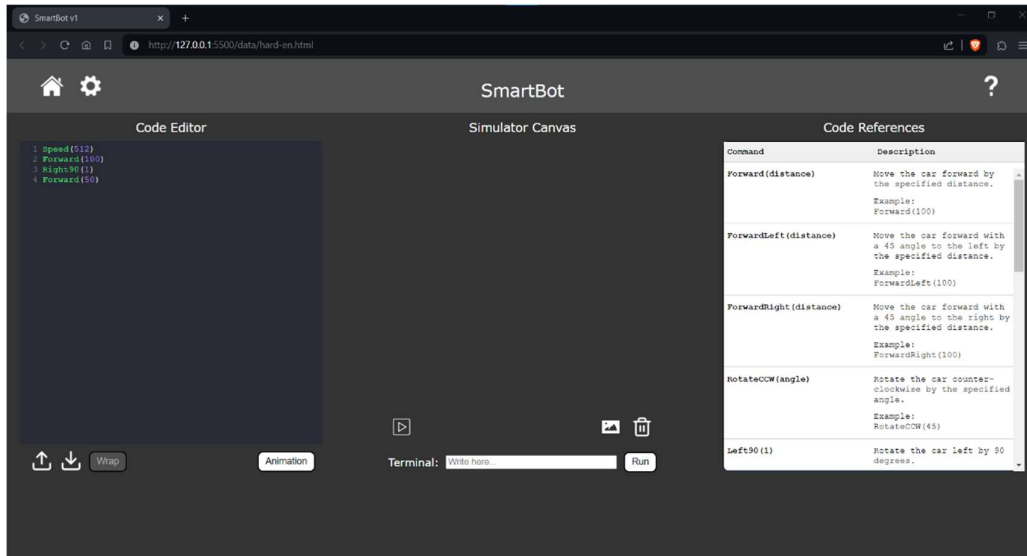
```
126 // Function to match the command pattern from codemirror and translate to tur
127 function interpretEspCommand(command, speed) {
128   var match = command.match(/(w|W)(([1-9]+))/);
129   if (match) {
130     var commandType = match[1];
131     var parameter = parseInt(match[2]); //The distance is parsed as par
132     switch (commandType) {
133       case 'forward':
134         case 'forwardleft':
135         case 'forwardright':
136         case 'RotateCCW':
137         case 'left90':
138         case 'RotateCW':
139         case 'Right90':
140         case 'backward':
141         case 'backwardleft':
142         case 'backwardright':
143         case 'Speed':
144           return [commandType, speed];
145         default:
146           console.log("Invalid command format:" + command);
147           codeVerifier = false;
148           return [];
149       }
150     } else {
151       console.log("Invalid command format:" + command);
152       codeVerifier = false;
153       return [];
154     }
155   }
156 // Function to handle the repetition logic
157 function repeatEspLogic(commandLines, startIndex, speed) {
158   var repeatLine = commandLines[startIndex];
159   var repeatMatch = repeatLine.match(/Repeat((\d+))/);
160   if (!repeatMatch) {
161     console.log("Invalid Repeat command format:", repeatLine);
162     return [];
163   }
164   // Function to handle the repetition logic
165   function repeatEspLogic(commandLines, startIndex, speed) {
166     var repeatLine = commandLines[startIndex];
167     var repeatMatch = repeatLine.match(/Repeat((\d+))/);
168     if (!repeatMatch) {
169       console.log("Invalid Repeat command format:", repeatLine);
170       return [];
171     }
172     var repeatCount = parseInt(repeatMatch[1]);
173     var interpretedCode = [];
174     for (var i = startIndex + 1; i < commandLines.length; i++) {
175       var line = commandLines[i];
176       if (line.trim() === '') {
177         break; // End of Repeat block
178       }
179       var interpretedCommand = interpretEspCommand(line, speed);
180       if (interpretedCommand != null) {
181         interpretedCode.push(interpretedCommand);
182       }
183     }
184     // Construct repeated commands
185     var repeatedCommands = [];
186     for (var j = 0; j < repeatCount; j++) {
187       repeatedCommands.push.apply(repeatedCommands, interpretedCode);
188     }
189     return repeatedCommands;
190   }
191   function runESP() {
192     var code = editor.getValue();
193     var codeVerifier = true;
194     // check for speed value
195     commandLines.forEach(function(line) {
196       if (line.trim().startsWith('Speed')) {
197         var match = line.match(/Speed((\d+))/);
198         // ...
199       }
200     });
201   }
202 }
```

Εικόνα 23: Στιγμιότυπο οθόνης από κώδικα ιστοσελίδας με μεταγλώττιση εντολών χρήστη σε εντολές οχήματος

Οι εντολές των πλήκτρων δημιουργήθηκαν χρησιμοποιώντας την βιβλιοθήκη jQuery και για λόγους διευκόλυνσης στον έλεγχο παρέμειναν στο κεντρικό αρχείο της ιστοσελίδας, ενώ οι λειτουργίες που επικαλούνται είναι αυτές του αρχείου που παρουσιάζεται στην Εικόνα 22 και Εικόνα 23. Η Εικόνα 24 παρουσιάζει την τελική μορφή που φέρει η ιστοσελίδα ελέγχου στο «δύσκολο» επίπεδο μετά την προσθήκη των παραπάνω λειτουργιών, καθώς επίσης του λεξικού εντολών στην τελευταία στήλη της ιστοσελίδας στα δεξιά. Παράλληλα, αξίζει να σημειωθεί ότι δημιουργήθηκε το ίδιο αρχείο και για τις δύο γλώσσες του προγράμματος,



μεταφράζοντας τα ονόματα και τις επεξηγήσεις και στα Ελληνικά αλλά και στα Αγγλικά.



Εικόνα 24: Στιγμιότυπο οθόνης από τελική ιστοσελίδα οχήματος στο «δύσκολο» επίπεδο

#### 4.4 Ανεπιτυχής λειτουργία

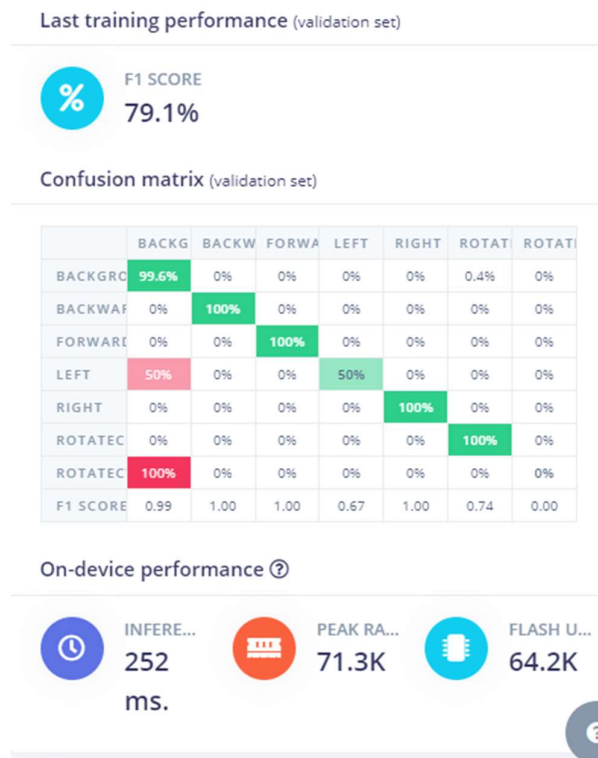
Στο κεφάλαιο αυτό και κατά την σχεδίαση της προσέγγισης στις επιμέρους αναβαθμίσεις που θα ήταν επιθυμητό να προστεθούν στον πηγαίο κώδικα του ρομποτικού οχήματος, έγινε αναφορά σε διάφορες πιθανές διαμορφώσεις που ίσως να μπορούσαν να εφαρμοστούν, μια εκ των οποίων είναι η επίτευξη ελέγχου μέσω οπτικής αναγνώρισης καρτών. Πραγματοποιήθηκαν αρκετές δοκιμές στο επίπεδο του εφικτού για την λειτουργία αυτή μέσω της προσθήκης του κώδικα συσχέτισης με τα μοντέλα οπτικής αναγνώρισης που παράγει η πλατφόρμα Edge Impulse (βλ. ενότητα 4.3.1.), παράγοντας μικτά αποτελέσματα.

Διατηρώντας την διαφάνεια στην έρευνα και εξερευνώντας τις δυνατότητες της πλακέτας, στην ενότητα αυτή παρουσιάζεται η ανεπιτυχής λειτουργία ελέγχου του οχήματος με κάρτες μέσω οπτικής αναγνώρισης. Από τις δοκιμές που έγιναν και αξίζει να αναφερθούν ως πηγή έμπνευσης για μελλοντικές αναβαθμίσεις, παρατηρήθηκε ότι η πλακέτα ESP32 – CAM με την έκδοση κάμερας 2MP OV2640, αν και μπορεί να υποστηρίξει μοντέλα οπτικής αναγνώρισης, δεν κατέχει τις απαραίτητες δυνατότητες για την αναγνώριση μικρού μεγέθους αντικειμένων ή

αντικειμένων που απαιτούν την επεξεργασία μικρού μεγέθους λεπτομερειών, κάτι που είναι αρκετά βασικό κατά την αναγνώριση καρτών.

Έτσι, ενώ η πλακέτα ESP32 – CAM θα μπορούσε να εντοπίσει αντικείμενα όπως μια τηλεόραση, ένα τραπέζι ή έναν υπολογιστή, μια γάτα ή ένα πουλί, και να χρησιμοποιήσει την εύρεση αυτών ή την σχετική τους θέση στην επιφάνεια που επιβλέπει η κάμερα για την δημιουργία διαφόρων προγραμμάτων και λειτουργιών, κάτι τόσο μικρό όσο μια κάρτα μεγέθους κάτω των 15 εκατοστών δεν μπορεί να αναγνωριστεί με αρκετή ακρίβεια ώστε να μπορεί να χρησιμοποιηθεί ως η κύρια μονάδα ελέγχου του οχήματος.

Κατά την μελέτη αυτής της λειτουργίας παρατηρήθηκε ότι ως ένα βαθμό εάν η κάρτα ξεπερνά σε μέγεθος τα 5 εκατοστά, έχει σαφή μορφή χωρίς πολλές λεπτομέρειες και η κάμερα λειτουργεί σε ασπρόμαυρη λήψη εικόνας, είναι δυνατή η επίτευξη αναγνώριση της κάρτας, ωστόσο με την προσθήκη πολλαπλών καρτών παρόμοιου σχεδιασμού που οι μικρές λεπτομέρειες κάνουν την διαφορά, η τελική ακρίβεια πέφτει δραματικά.

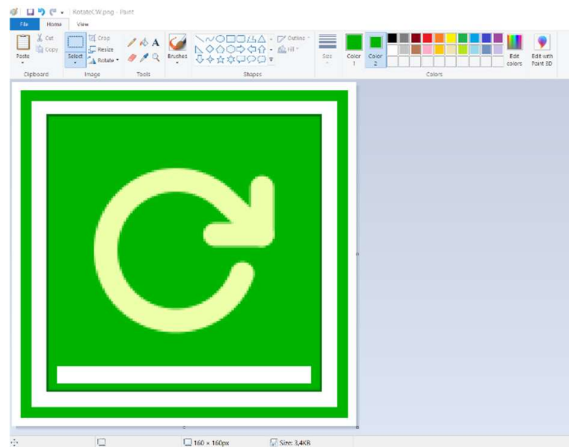


Εικόνα 25: Στιγμιότυπο οθόνης από το γράφημα σύγχυσης του εκπαιδευμένου μοντέλου (αρχικά αποτελέσματα)

Στην Εικόνα 25 γίνεται παρουσίαση των πρώτων αποτελεσμάτων από το μοντέλο οπτικής αναγνώρισης που πραγματοποιήθηκε, όπου τα αρχικά αποτελέσματα του μοντέλου οπτικής αναγνώρισης καρτών παρουσιάζει τα ποσοστά σύγχυσης του μοντέλου ανάμεσα στα αναγνωρίσιμα αντικείμενα. Η πλήρη διαδικασία που ακολουθήθηκε εμπεριείχε τα βήματα που αναλύονται παρακάτω, με τα βήματα τρία έως έξι, να επαναλαμβάνονται αρκετές φορές στην προσπάθεια επίτευξης ενός πλήρως λειτουργικού αποτελέσματος.

### Βήμα 1<sup>ο</sup>

Η αρχή του εγχειρήματος ελέγχου του ρομποτικού συστήματος με κάρτες που θα παίρνουν την θέση εντολών γίνεται με τον σχεδιασμό των καρτών στο πρόγραμμα της Microsoft MS Paint, παίρνοντας ως βάση τις εικόνες εντολών του «μέτριου» επιπέδου δυσκολίας και μορφοποιώντας τις λεπτομέρειες περιμετρικά του σήματος για την διευκόλυνση αναγνώρισης. Το σχέδιο της κάρτας πέρασε από αρκετές παραλλαγές, με την πιο επιτυχημένη μορφοποίηση να παρουσιάζεται στην Εικόνα 26.



Εικόνα 26: Ενδεικτικό στιγμιότυπο οθόνης κατά την σχεδίαση μιας από τις κάρτες ελέγχου

### Βήμα 2<sup>ο</sup>

Μετά την σχεδίαση των καρτών ηλεκτρονικά και την εκτύπωση τους, είναι απαραίτητη η δημιουργία ενός χώρου για την συλλογή εικόνων που θα χρησιμοποιηθούν στην εκπαίδευση του μοντέλου οπτικής αναγνώρισης. Για την επίτευξη καλύτερων αποτελεσμάτων είναι σημαντικό να χρησιμοποιηθεί ένα καθαρό

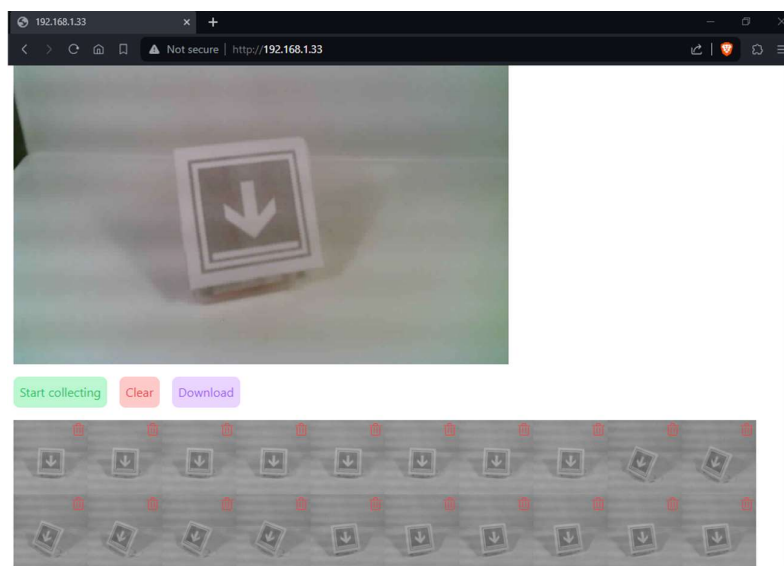
και μονόχρωμο περιβάλλον, αρκετά καλά φωτισμένο για την λήψη των εικόνων χωρίς να υπάρχει η παραμικρή σύγχυση του αντικειμένου που πρέπει να αναγνωριστεί από το ελεγχόμενο περιβάλλον της φωτογράφισης. Στην Εικόνα 27 γίνεται ανάδειξη του αυτοσχέδιου φόντου για την φωτογράφιση των καρτών που χρησιμοποιήθηκε στην μελέτη αυτή.



Εικόνα 27: Εικόνες του πάγκου εργασίας κατά την λήψη φωτογραφιών μέσω της πλακέτας ESP32 για το μοντέλο εκπαίδευσης οπτικής αναγνώρισης

### Βήμα 3<sup>ο</sup>

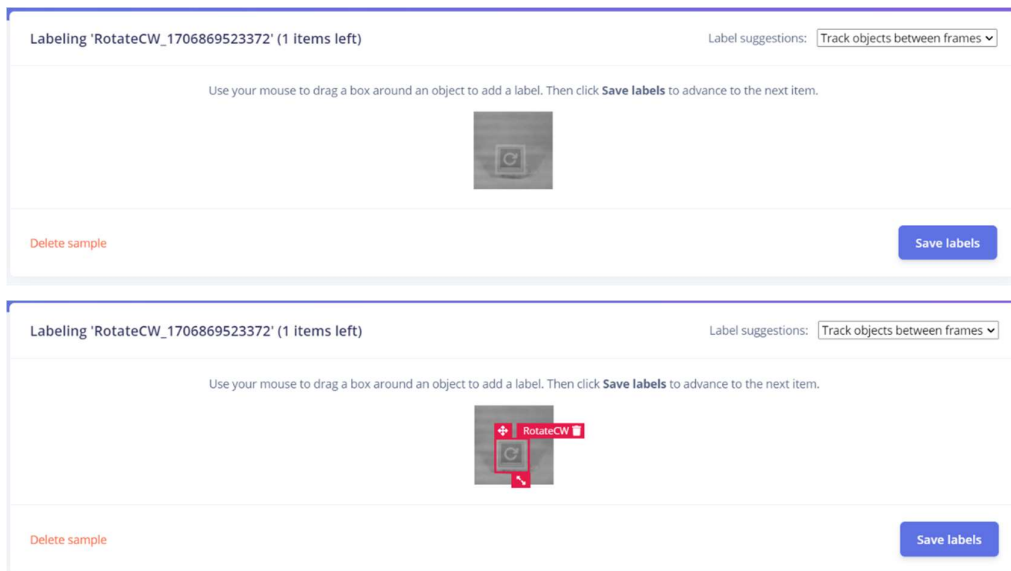
Το πρόγραμμα που χρησιμοποιήθηκε στην πλακέτα ESP32 CAM για την συλλογή εικόνων από κάθε κάρτα - εντολή που θα πρέπει να μπορεί η κάμερα να αναγνωρίσει επιλέχθηκε από σχετική βιβλιοθήκη ανοιχτού κώδικα του EloquentArduino, 2022. Στην βιβλιοθήκη αυτή παρουσιάζονται κώδικες παραδείγματα χρησιμοποιώντας την πλακέτα ESP32 για την δημιουργία μοντέλων οπτικής αναγνώρισης, με το παράδειγμα 26 να φέρει τον απαραίτητο κώδικα για την συλλογή των εικόνων.



Εικόνα 28: Ιστοσελίδα ελέγχου πλακέτας ESP32 και του προγράμματος `26_collect_images.ino` του EloquentArduino

#### Βήμα 4°

Έχοντας συλλέξει αρκετά δείγματα από κάθε αντικείμενο που θα πρέπει το μοντέλο να αναγνωρίσει, καθώς επίσης από το περιβάλλον της φωτογράφισης χωρίς κανέναν αντικείμενο για να μπορεί το μοντέλο να ξεχωρίσει τα αντικείμενα από το περιβάλλοντα χώρο, σειρά έχει η προσθήκη των εικόνων στην πλατφόρμα Edge Impulse. Κατά την εισαγωγή δεδομένων στην πλατφόρμα, ο χρήστης θα πρέπει να επεξεργαστεί κάθε εικόνα με στόχο την επιλογή και την ονομασία των αντικειμένων σε κάθε εικόνα. Οι επιλογές αυτές και τα ονόματα που θα επιλέξει ο χρήστης σε αυτό το σημείο θα χρησιμοποιηθούν ως βάση από την πλατφόρμα για την εκπαίδευση του μοντέλου οπτικής αναγνώρισης και της εύρεσης πληροφοριών που θα το βοηθούν να βρει τα επιθυμητά αντικείμενα.



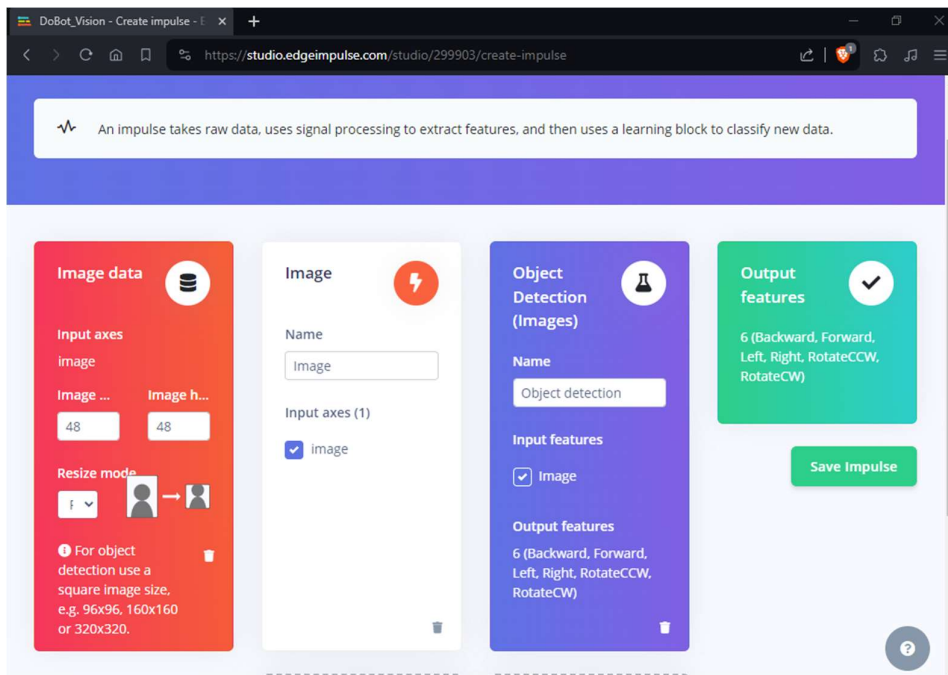
Εικόνα 29: Στιγμιότυπα οθόνης από την επιλογή της κάρτας περιστροφής δεξιόστροφα στην πλατφόρμα Edge Impulse

Οι εικόνες στην συνέχεια διαχωρίζονται σε δύο ομάδες, τα δεδομένα εκπαίδευσης του μοντέλου, μέσω των οποίων το μοντέλο θα προσπαθήσει να βρει πληροφορίες που θα του επιτρέπουν να αναγνωρίζει το κάθε εικονιζόμενο αντικείμενο, και τα δεδομένα δοκιμής, τα οποία θα χρησιμοποιεί σε κάθε κύκλο εκπαίδευσης της πλατφόρμας για να βρίσκει το επίπεδο επιτυχίας στην εύρεση και ορθή αναγνώριση των αντικειμένων. Τον αριθμό των εικόνων που θα πρέπει να εμπεριέχει η κάθε ομάδα ορίζεται προσεγγιστικά στο 80% για τα δεδομένα εκπαίδευσης και 20% για τα δεδομένα δοκιμής.

## Βήμα 5°

Πέραν την προσθήκης δεδομένων στην πλατφόρμα εκπαίδευσης θα πρέπει να γίνει η απαραίτητη παραμετροποίηση του μοντέλο οπτικής αναγνώρισης κατά την σχεδίαση του εκπαιδευτικού μηχανισμού. Μετά την αρχική ανάλυση των εικόνων για την εύρεση των πληροφοριών που θα βοηθήσουν το μοντέλο να αναγνωρίσει τις διάφορες κάρτες ελέγχου, γίνεται επεξεργασία των ρυθμίσεων που θα φέρει ο τελικός κώδικας όσο θα εκτελείται το μοντέλο αναγνώρισης.

Πιο συγκεκριμένα γίνεται επιλογή του μεγέθους των εικόνων που θα ελέγχει το σύστημα, ορίζοντας τις εικόνες σε μέγεθος 48px x 48px, μετατρέποντας τις εικόνες σε αυτό το μέγεθος συμπιέζοντας όλες τις πλευρές τους. Ταυτόχρονα επιλέγεται ο σκοπός του μοντέλου (λ.χ. αναγνώριση αντικειμένων) και εισαγωγή των ονομάτων από τα αντικείμενα του βήματος 4 τα οποία θα πρέπει να επιστρέφει το μοντέλο αναγνώρισης όταν εντοπίζει οποιοδήποτε από αυτά. Οι επιλογές αυτές πλην του μεγέθους των εικόνων παραμένουν σταθερά οι ίδιες, ενώ το μέγεθος της εικόνας που θα λαμβάνεται και θα ελέγχεται για την εύρεση του αντικειμένου φέρει άμεση σχέση με την χαμηλή χωρητικότητα της μνήμης της πλακέτας ESP32 και του επεξεργαστή S2 που χρησιμοποιεί.



Εικόνα 30: Στιγμιότυπο οθόνης από την πλατφόρμα Edge Impulse κατά την προσαρμογή ρυθμίσεων

Αν και είναι εφικτό να τρέξουν προγράμματα με μέγεθος έως και 96px x 96px κάτι τέτοιο θα πρόσθετε αρκετά μεγάλη καθυστέρηση στο σύστημα, κάνοντας την εύρεση αντικειμένων πρακτικά ασύγχρονη και δύσχρηστη για την δημιουργία οποιασδήποτε λειτουργίας για το όχημα αναφοράς. Το μέγιστο μέγεθος που χρησιμοποιήθηκε από την παρούσα μελέτη που να μην επέφερε πολύ μεγάλη καθυστέρηση στην αναγνώριση ήταν αυτό των 64px x 64px αν και στην περίπτωση αυτή το πρόγραμμα είχε μεγαλύτερη δυσκολία στον διαχωρισμό των καρτών μεταξύ τους, έχοντας περισσότερα δεδομένα να επεξεργαστεί κάθε φορά.

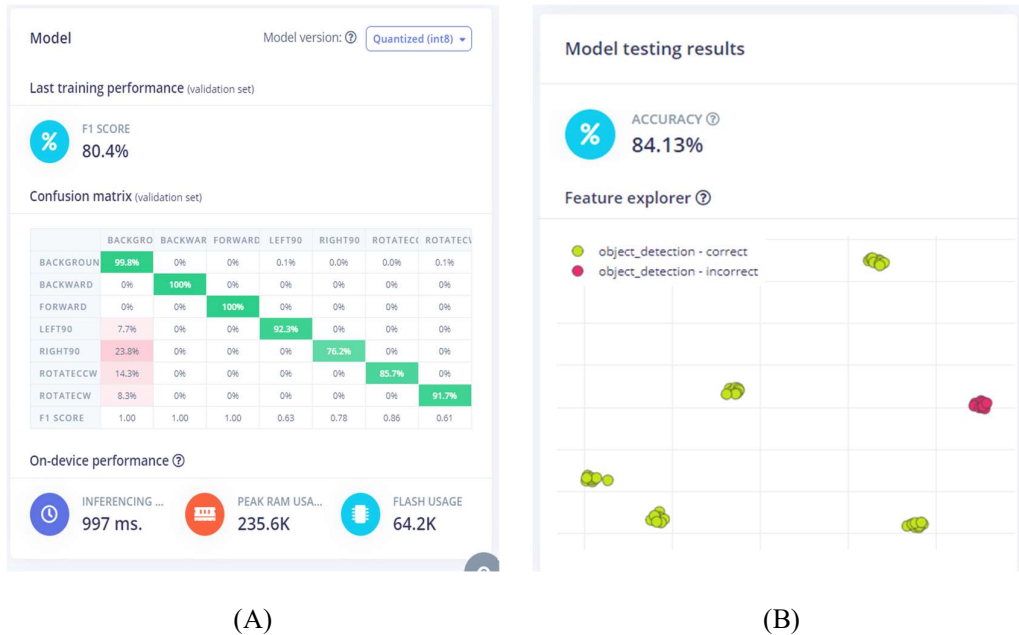
Τόσο η επιλογή του μεγέθους των εικόνων όσο και της χρήσης ασπρόμαυρης λήψης για την εκτέλεση του μοντέλου αναγνώρισης γίνονται με γνώμονα την ισχύ του επεξεργαστή της προεπιλεγμένης πλακέτας, συνεπώς με την επιλογή της καινούριας γενιάς επεξεργαστών S3, σαφώς οι επιλογές αυτές μπορούν να προσαρμοστούν και πιθανότατα να επιτρέψουν την ορθή λειτουργία αυτού του εγχειρήματος. Ωστόσο, οι νέες πλακέτες δεν είναι άμεσα διαθέσιμες στα χρονικά περιθώρια της παρούσας διπλωματικής εργασίας και για τον λόγο αυτό δεν έγινε επιλογή ανάλυσης τους.

## Βήμα 6<sup>ο</sup>

Επιλέγοντας 30 με 50 κύκλους εκπαίδευσης του μοντέλου, με ρυθμό εκπαίδευσης 0,001 έως 0,005, τα αποτελέσματα του εκπαιδευόμενου μοντέλου δεν ξεπερνούσαν ποτέ το 0.85 βαθμό βεβαιότητας όταν αναγνωρίζονταν η ύπαρξη κάρτας μπροστά από την κάμερα. Ενώ η σύγκριση καρτών μεταξύ τους δεν ξεπέρασε ποτέ το 0.6 βαθμό βεβαιότητας σε πραγματικές δοκιμές του παραγόμενου από την πλατφόρμα κώδικα. Σε αυτό το βήμα τα αποτελέσματα συγκρίνονται με τις προηγούμενες δοκιμές και μεταβάλλοντας τις παραμέτρους εκπαίδευσης επαναλαμβάνονται ορισμένα από τα βήματα για την επίτευξη καλύτερου τελικού αποτελέσματος.

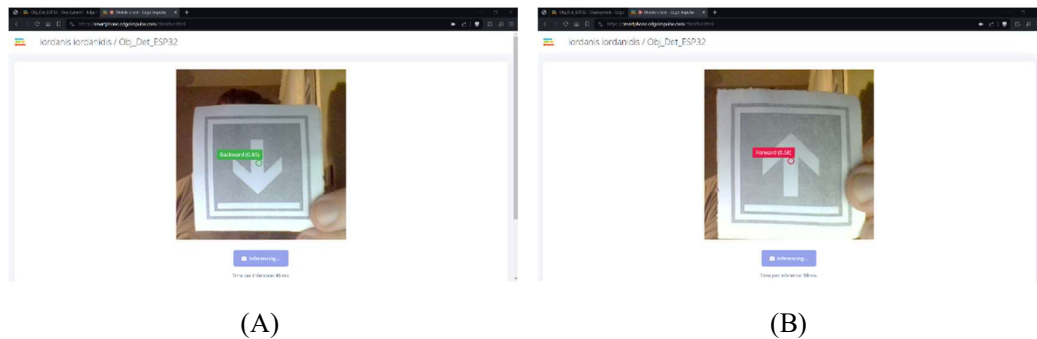
Ο έλεγχος μέσω καρτών κατέστη εφικτός μόνο στην περίπτωση όπου το παραγόμενο για την πλακέτα ESP32 CAM πρόγραμμα έτρεχε σε σύνδεση με εξωτερική κάμερα για υπολογιστές, χρησιμοποιώντας σύνδεση σε αυτή μέσω Wi-Fi ή στην περίπτωση όπου εκτελούνταν τοπικά με την χρήση της ίδιας της πλατφόρμας Edge Impulse. Τα αποτελέσματα και στις δύο αυτές περιπτώσεις έφεραν αποτελέσματα άνω του 80% στα γραφήματα σύγκρισης των καρτών αλλά και σε δοκιμαστικούς ελέγχους, είτε χρησιμοποιούνταν η εξωτερική κάμερα είτε σε ήδη αποθηκευμένες εικόνες που είχε

τραβήξει νωρίτερα η πλακέτα ESP32 CAM αντίστοιχα. Τα αποτελέσματα αυτά παρουσιάζονται στην Εικόνα 31.



Εικόνα 31: Αποτελέσματα δοκιμών του μοντέλου αναγνώρισης με χρήση εξωτερικής κάμερας (A) ή της πλατφόρμας Edge Impulse σε τοπικό υπολογιστή (B)

Ακόμα και σε αυτές τις περιπτώσεις τα αποτελέσματα δεν φέρουν βεβαιότητα σε επιθυμητά επίπεδα (λ.χ. άνω του 0.9), ωστόσο τόσο η ταχύτητα όσο και η έλλειψη μεγάλης σύγχυσης μεταξύ καρτών και περιβάλλοντος τα θέτουν ικανά στον χειρισμό του οχήματος κοντά στα επιθυμητά επίπεδα. Στην Εικόνα 32 έγινε προσθήκη στιγμιότυπων από την οθόνη του τελικού χρήστη κατά την παρουσίαση δύο καρτών κίνησης με αντίθετη φορά η μία από την άλλη και αρκετή ομοιότητα ανάμεσα στα αντικείμενα.



Εικόνα 32: Στιγμιότυπο οθόνης κατά την εκτέλεση του κώδικα στην πλατφόρμα Edge Impulse σε τοπικό υπολογιστή όπου (A) 0.85 βεβαιότητα κάρτα κίνησης πίσω και (B) 0.58 βεβαιότητα κάρτα κίνησης μπροστά.



## 5 ΣΥΖΗΤΗΣΗ

### 5.1 Συζήτηση

Η διαδικασία εξέλιξης και βελτίωσης ενός υφιστάμενου ρομποτικού συστήματος που θα χρησιμοποιηθεί στην πρωτοβάθμια εκπαίδευση, βασίζεται στην προσέγγιση ανάστροφης μηχανικής, μέσω της οποίας αναλύεται ο ακριβής τρόπος λειτουργίας του ρομποτικού οχήματος πίσω από τον χειρισμό του μέσω ιστοσελίδας. Παράλληλα, χρησιμοποιούνται δύο κύκλοι βελτιώσεων που αντιστοιχούν σε τεχνικές καταγισμού ιδεών για την συλλογή πιθανών βελτιώσεων και την υλοποίηση όσων τελικά επιλεχθούν να προστεθούν στο σύστημα αναφοράς.

Τα αποτελέσματα της παρούσας διπλωματικής εργασίας συνοψίζονται μέσω της παραγωγής πηγαίου κώδικα για το σύνολο του ρομποτικού συστήματος αναφοράς και των προστιθέμενων λειτουργιών που προκύπτουν από την παραγωγή αυτή. Ο κώδικας προσθέτει μερικές νέες λειτουργίες, ενώ παράλληλα βελτιώνει τις υπάρχουσες δυνατότητες του συστήματος, με στόχο την επαύξηση της εκπαιδευτικής εμπειρίας για μαθητές της πρωτοβάθμιας εκπαίδευσης. Το σύστημα που επιλέγεται, αναφέρεται ως παράδειγμα σε αρκετές μελέτες σχετικά με την επίδραση που επιφέρουν παρόμοια συστήματα στον εκπαιδευτικό κλάδο τόσο στην πρωτοβάθμια εκπαίδευση όσο και σε μικρότερες ηλικιακές ομάδες. Αν και δεν υπάρχει άμεσο μέτρο σύγκρισης των αποτελεσμάτων που φέρουν οι αλλαγές στο σύστημα αναφοράς, η αύξηση των δυνατοτήτων του οχήματος στοχεύει στην ενίσχυση αυτής της συσχέτισης, προσφέροντας περισσότερες δυνατότητες εκπαιδευτικής δράσης.

Παρόλες τις βελτιώσεις που επιλέγονται να γίνουν και των προσπαθειών για την επιτυχή προσθήκη τους στο σύστημα, ορισμένα προβλήματα προκύπτουν κατά την διάρκεια σχεδίασης και εκτέλεσης των βελτιώσεων, με το κυριότερο όλων τον περιορισμό σε επεξεργαστική ισχύ που φέρει η πλακέτα ESP8266. Σε κάποιες λειτουργίες ο διαμοιρασμός των λειτουργιών με την δευτερεύουσα προτεινόμενη πλακέτα ή μέσω αλλαγής της κύριας πλακέτας με κάποια καλύτερη, μπορούν να επιτρέψουν την βελτίωση του συστήματος. Ωστόσο, δεν είναι επιτυχημένες όλες οι απόπειρες προσθήκης νέων λειτουργιών στο σύστημα αναφοράς, με την παρουσίαση της ανεπιτυχούς προσθήκης ελέγχου μέσω οπτικής αναγνώρισης καρτών να

παρουσιάζεται ως ένδειξη διαφάνειας (βλ. ενότητα 4.4) και αντιμετώπισης των προκλήσεων κατά την διάρκεια της έρευνας.

## **5.2 Μελλοντική Έρευνα**

Με βάση τα μέχρι τώρα αποτελέσματα της διπλωματικής εργασίας και αντλώντας έμπνευση από τον πίνακα 1 (βλ. ενότητα 4.1) προκύπτουν κάποιες επιλογές οι οποίες δεν επιλέχθηκαν να αναλυθούν ή να εφαρμοστούν στην παρούσα φάση του ρομποτικού συστήματος αναφοράς, ωστόσο θα μπορούσαν να ερευνηθούν περαιτέρω σε μελλοντικό επίπεδο.

### **5.2.1 Βιβλιοθήκη Wi-Fi Manager**

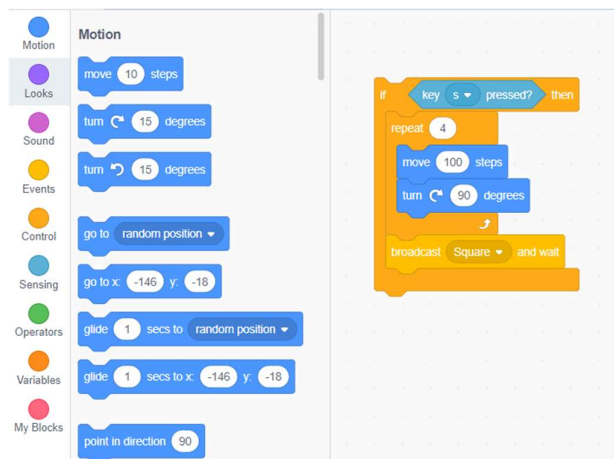
Από τις πρώτες επιλογές του πίνακα 1 που διερευνήθηκαν και εν τέλει δεν επιλέχθηκε να αναλυθούν και να εφαρμοστούν είναι αυτή της χρήσης Wi-Fi Manager. Το αρχικό σύστημα φέρει εσωτερικά πρωτόκολλο Wi-Fi για την δημιουργία αυτόνομου τοπικού δικτύου για την σύνδεση των χρηστών στο όχημα, όπου με την προσθήκη της βιβλιοθήκης Wi-Fi Manager, θα είναι σε θέση ο χρήστης να επιλέξει μεταξύ αυτόνομου τοπικού δικτύου ή σύνδεση σε υπάρχον δίκτυο Wi-Fi. Εισάγοντας την βιβλιοθήκη *WiFiManager.h* στο σύστημα αναφοράς, η παραπάνω λειτουργία μπορεί να επιτευχθεί, ενώ παράλληλα, η εισαγωγή κωδικού αλλά και του ονόματος για το δίκτυο που θα συνδεθεί το όχημα θα μπορεί να γίνεται τοπικά μέσω της βιβλιοθήκης *WiFiManager.h*, χωρίς την ανάγκη επεξεργασίας του πηγαίου κώδικα και της εισαγωγής των στοιχείων του δικτύου μέσω αυτού.

Στο παρόν επίπεδο του συστήματος αναφοράς δεν συντρέχουν λόγοι χρήσης τοπικού δικτύου για την σύνδεση του χρήστη στο όχημα, αλλά ούτε χρειάζεται σύνδεση στο διαδίκτυο για οποιαδήποτε από τις λειτουργίες του. Υπό συνθήκες όπου η πλακέτα θα φέρει λειτουργίες που την υποχρεώνουν να επικοινωνεί με τοπικά δίκτυα για την πρόσβαση στο διαδίκτυο, ή στην περίπτωση που θα χρησιμοποιείται εξ ολοκλήρου μόνο σε χώρους που έχουν πρόσβαση σε τοπικό δίκτυο Wi-Fi, τότε ίσως η επιλογή της βιβλιοθήκης αυτής θα πρέπει να εξεταστεί, καθώς επιτρέπει μεταξύ άλλων την κλήση αρκετών από των βιβλιοθηκών που χρησιμοποιούνται για τις διάφορες λειτουργίες ζωντανά μέσω του διαδικτύου αντί να αποθηκεύονται και να καλούνται τοπικά, ελαφραίνοντας το συνολικό πρόγραμμα.

## 5.2.2 Σύνδεση με Γλώσσα Scratch

Μια ακόμα θεωρητική ιδέα η οποία δεν διερευνήθηκε σε μεταγενέστερο στάδιο είναι αυτή της χρήσης γλώσσας προγραμματισμού Scratch, που είναι αρκετά φιλική προς τις παιδικές ηλικίες καθώς φέρει την μορφή πλακιδίων που συνδυαστικά δημιουργούν έναν κώδικα. Η γλώσσα αυτή επιτρέπει την δημιουργία πρόσθετων (plug in) συνδέσεων, κάνοντας χρήση προσαρμοσμένου κώδικα βιβλιοθήκης, που θα μπορούσε να σχεδιαστεί με βάση τις εντολές που φέρει ήδη το ρομποτικό όχημα, επιτρέποντας τον προγραμματισμό του με αυτόν τον τρόπο.

Μια τέτοια προσθήκη ωστόσο χρειάζεται τόσο την δημιουργία προσωπικών εντολών – πλακιδίων υπό την μορφή βιβλιοθήκης για την χρήση σε μεταγλωττιστές της γλώσσας Scratch, όσο και του προγράμματος από την πλευρά της πλακέτας ESP8266 που θα επικοινωνεί με τον μεταγλωττιστή, και την plug in σύνδεση που θα μεταφράζει τα πλακίδια σε εντολές ελέγχου που θα καταλαβαίνει το πρόγραμμα.



Εικόνα 33: Παράδειγμα γλώσσας προγραμματισμού Scratch

Η επιλογή αυτή απορρίφθηκε διότι ομοιάζει αρκετά στο ήδη υπάρχον σύστημα ελέγχου του «μεσαίου» επιπέδου δυσκολίας χειρισμού του οχήματος, με την διαφορά ότι αυτό έχει δημιουργηθεί τοπικά για τις ανάγκες του οχήματος και μόνο, αποφεύγοντας μέρος των επιπλέον δυσκολιών που θα έφερε ένα τέτοιο εγχείρημα. Ωστόσο, η γλώσσα προγραμματισμού Scratch προσφέρει ένα πλήθος λογικών λειτουργιών που θα μπορούσαν να χρησιμοποιηθούν σε συνεργασία με το πρόγραμμα του οχήματος αντί του «μεσαίου» επιπέδου ελέγχου, κάνοντας αυτήν την επιλογή αρκετά δελεαστική για μελλοντική έρευνα.

### **5.2.3 ESP32 Τηλεχειριστήριο**

Στις δυνατότητες επεκτάσεων και αναβαθμίσεων του ρομποτικού συστήματος και συνδυαστικά με την προσθήκη του πρωτοκόλλου ESP – NOW και την χρήση της πλακέτας ESP32, είναι και αυτή της δημιουργίας μιας δευτερεύουσας συσκευής που θα φέρει την μορφή τηλεχειριστήριου για τον έλεγχο του οχήματος. Εάν μάλιστα γίνει επιλογή της πλακέτας ESP32 – CAM ή ESP32 – EYE, οι επιπλέον δυνατότητες που εμπεριέχουν μέσω των ενσωματωμένων καμερών ή του μικροφώνου στην περίπτωση του δεύτερου, μπορούν να αποτελέσουν βάση για ένα πλήθος νέων τρόπων ελέγχου του οχήματος (λ.χ. έλεγχος με φωνητικές εντολές, έλεγχος μέσω χειρονομιών). Στο επίπεδο της διπλωματικής εργασίας, η επιλογή αυτή δεν εξερευνήθηκε περαιτέρω, αν και προτείνεται ως ιδέα για μελλοντική μελέτη.

### **5.2.4 Χρήση Bootstrap UI και Bootstrap θεμάτων**

Μια ακόμα πρόταση για το υπάρχον σύστημα που στόχο έχει την βελτίωση της ιστοσελίδας στο σύνολο της, είναι η επαναδημιουργία της μέσω του εργαλείου ανάπτυξης ιστοσελίδων και διεπαφών Bootstrap και την χρήση των θεμάτων Bootstrap για τον σχεδιασμό των σελίδων. Μια τέτοια αλλαγή προσφέρει ποικιλία θεμάτων που θα μπορούν να μεταβάλλονται βάση επιλογής του χρήστη για την μορφοποίηση της σελίδας, καθώς επίσης θα μπορούν να παραμετροποιηθούν με βάση την οθόνη που προβάλλεται η ιστοσελίδα, τα διαφορά αντικείμενα που παρουσιάζονται σε αυτήν. Η συλλογή θεμάτων εμπεριέχει λειτουργίες μεταβολής παραμέτρων σχεδιασμού βάση του μεγέθους αλλά και του είδους της οθόνης στην οποία προβάλετε η ιστοσελίδα, φέρνοντας την σελίδα στα μέτρα της όποια οθόνης χρησιμοποιηθεί, ασχέτως της οθόνης στην οποία σχεδιάστηκε το σύστημα αρχικά.

## 6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η διπλωματική εργασία εστιάζεται στην εξέλιξη και τη βελτίωση ενός ρομποτικού συστήματος που προορίζεται για χρήση στην πρωτοβάθμια εκπαίδευση. Με την προσέγγιση της ανάστροφης μηχανικής, αναλύθηκε η λειτουργία του ρομποτικού οχήματος και προτάθηκαν βελτιώσεις που επικεντρώνονται στην ενίσχυση της εκπαιδευτικής εμπειρίας των μαθητών.

Μέσω δύο κύκλων βελτιώσεων, συλλέχθηκαν ιδέες και υλοποιήθηκαν αλλαγές στο ρομποτικό σύστημα, με τον παραγόμενο πηγαίο κώδικα να περιλαμβάνει νέες λειτουργίες που επιτρέπουν περισσότερες δυνατότητες εκπαιδευτικής δράσης. Παράλληλα, έγινε προσπάθεια αντιμετώπισης προβλημάτων που προέκυψαν κατά την διάρκεια της επεξεργασίας του πηγαίου κώδικα, όπως οι περιορισμοί σε επεξεργαστική ισχύ της πλακέτας ESP8266 και ο περιορισμένος χώρος αποθήκευσης στην μνήμη της πλακέτας.

Οι περισσότερες προσθήκες κατέστησαν επιτυχημένες, εκτός από την προσθήκη ελέγχου μέσω οπτικής αναγνώρισης καρτών. Η λειτουργία αυτή αναδείχθηκε ως ανεπιτυχής καθώς το παραγόμενο από την πλατφόρμα Edge Impulse πρόγραμμα δεν ήταν σε θέση να λειτουργήσει όπως προβλέπεται όσο χρησιμοποιούνταν ο συνδυασμός της πλακέτας AI Thinker ESP32 CAM που επιλέχθηκε με την προεγκατεστημένη κάμερα OV2640.

Τέλος, προκύπτει η ανάγκη για περαιτέρω έρευνα στο κατά πόσο οι παραπάνω επιλεγμένες μεταβολές του συστήματος επηρεάζουν την διδασκαλία εκπαιδευτικής ρομποτικής, καθώς και ανάπτυξη του συστήματος αναφοράς προκειμένου να εξελιχθεί περισσότερο το σύστημα, καλύπτοντας έτσι μεγαλύτερο εύρος δυνατοτήτων.

## 7 ΒΙΒΛΙΟΓΡΑΦΙΑ

Chatzopoulos, A., Papadakis, S., Kalogiannakis, M., & Papoutsidakis, M. (2022). A Novel, Modular Robot for Educational Robotics Developed Using Action Research Evaluated on Technology Acceptance Model. *Education Sciences*.

EloquentArduino. (2024). EloquentEsp32cam. Retrieved from GitHub via <https://github.com/eloquentarduino/EloquentEsp32cam> (12/02/2024)

Moomaw, S. (2012). STEM Begins in the Early Years. *School Science and Mathematics*, 112: 57-58.

Οργανισμός Εκπαιδευτικής Ρομποτικής, Επιστήμης, Τεχνολογίας & Μαθηματικών. (2024). Τι είναι STEM. Διαθέσιμο στον δικτυακό τόπο: <https://stem.edu.gr/τί-είναι-stem/> (14/01/2024)

Stone-MacDonald, A., Wendell, K., Douglass, A., & Love, M. L. (2015). *Engaging young engineers: Teaching problem-solving skills through STEM*. University of Massachusetts, Boston

## 8 ΠΑΡΑΡΤΗΜΑΤΑ

Την παρούσα διπλωματική εργασία συνοδεύουν αρχεία πηγαίου κώδικα που δημιουργήθηκαν για τις ανάγκες ολοκλήρωσης, της βελτίωσης του μη επανδρωμένου εκπαιδευτικού χερσαίου οχήματος για την πρωτοβάθμια εκπαίδευση. Το συνολικό έργο είναι αποθηκευμένο σε έναν κεντρικό φάκελο ο οποίος διαχωρίζεται σε υποφακέλους και αρχεία όπως ακολουθεί:

data (File)	cm (File)	clike.min.js
		codemirror.min.css
		codemirror.min.js
		Dracula.min.css
		jquery.min.js
		python.min.js
	css (File)	blocks.css
		blocks_old.css
		template.css
		template_old_30-4-20.css
		template-old-2023.css
	examples (File)	desktop.ini
		spiral.js
		tiny_circle.js
		triangle.js
	icons (File)	Back_to_center.png
		Backward.png
		BackwardLeft.png
		BackwardRight.png
		Beep.png
		Beep_blue_white.png
		Beep_yalloew_blue.png
		Beep_yalloew_green.png
		Circle.png
		Delay.png
		Delay_blue.png
		Forward.png

		ForwardLeft.png
		ForwardRight.png
		Hide_turtle.png
		Home.png
		Left.png
		Left90.png
		Left90_old.png
		Pen_Down.png
		Pen_Up.png
		Repeat.png
		Right.png
		Right.png
		RotateCCW.png
		RotateCW.png
		Show_turtle.png
		Sound.png
		Sound_blue.png
		Sound_blue_white.png
		Sound_yellow_green.png
		Speed.png
		Seed_blue.png
		Speed_old.png
		Start.png
		Stop.png
	images (File)	Delete-icon.png
		Easy.png
		Examples-icon.png
		Greek-Flag.png
		Hard.png
		Help-icon.png
		Home-icon.png
		Image-icon.png
		Load-icon.png
		Medium.png



	Run-icon.png
	Save-icon.png
	Settings-icon.png
	UK-Flag.png
js (File)	blocks.js
	blocks_old.js
	blocks_medium.js
	custom-cmd.js
	drag.js
	drag_old.js
	file.js
	file_old.js
	menu.js
	menu_old.js
	menu-medium.js
	socket.js
	triangle_old.js
	turtle.js
	turtle_old.js
	turtle-medium.js
	util.js
	util_old.js
lib (File)	dragula.min.css
	dragula.min.js
sk (File)	debugger.js
	skulpt.min.js
	skulpt-stdlib.js
easy-en.html	
easy-gr.html	
favicon.ico	
hard-en.html	
hard-gr.html	
index.html	
indexBloc.html	

	index-en.html
	medium-en.html
	medium-gr.html
ESP32 (File)	espnow_esp32_sender.ino
	espnow_esp32_sender_ei.ino
espnow_esp8266_receiver.ino	
mDNS_dubot.ino	
Movement_Functions.ino	
notes.txt	
SmartBot_Language.ino	
SmartBot_v5.ino	
SPIFFS_SPI_Flash_System.ino	
to_delete.ino	
WebServer.ino	
WebSocket.ino	
WebSocket_Send_Data.ino	
Z_Instructions_and_Help.ino	

Από την παραπάνω δομή του έργου, μερικά από τα αρχεία έχουν δημιουργηθεί ή υποστεί επεξεργασία από τον συγγραφέα της διπλωματικής, ενώ τα υπόλοιπα αποτελούν μέρος του αρχικού συστήματος ή είναι βιβλιοθήκες ανοιχτού κώδικα χωρίς κάποια παρέμβαση. Τα αρχεία που περιέχουν έργο του συγγραφέα της διπλωματικής θα καταγραφούν πλήρως στο παρών παράρτημα.

## Project/Smartbot\_v5.ino

```
// Libraries to include
#include <ESP8266WiFi.h> // Include the Wi-Fi library
#include <WiFiClient.h>
#include <ESP8266WebServer.h> // Include WebServer library
#include <FS.h> // Include the SPIFFS ("data"
folder upload to Flash Ram) library
#include <WebSocketsServer.h> // Include the WebSocket
library

// #include <SocketIOclient.h>
// #include <WebSockets.h>
// #include <WebSocketsClient.h>
// #include <WebSocketsServer.h>

// Constants & Variables
const char *ssid = "SmartBot"; // The name of the Wi-Fi network
that will be create

const char *password = "SmartBot"; // The password required to
connect to it, leave blank for an open network

const int MRp = 5; // Motor Right Plus pin
const int MRm = 4; // Motor Right Minus pin
const int MLp = 14; // Motor Left Plus pin
const int MLm = 12; // Motor Left Minus pin
const int Button_pin = 10; // Button pin
int Button_value = 0; // Initial Button read value
int Button_state = 0; // Button state
const int LDR_pin = A0; // LDR
int LDR_value = 0; // Initial LDR read value
int LDR_pr_value = 0; // Previous LDR read value
int robotSpeed = 500; // Initial Robot Speed (0-1023,
1023 is the max speed)
int robotMoveNum = 1; // Initial Robot Move (1 is for
FORWARD)

// The following variables are for time-software interrupts
unsigned long t; // Use t to store millis()
```

```

unsigned long last_10sec = 0;           // Use this is as 10sec time
interval

unsigned long last_3sec = 0;           // Use this is as 3sec time
interval

unsigned long last_1sec = 0;           // Use this is as 1sec time
interval

unsigned long last_05sec = 0;         // Use this is as 0.5 sec time
interval

unsigned long last_03sec = 0;         // Use this is as 0.3 sec time
interval

// unsigned int counter = 0;

// The following are functions

bool handleFileRead(String path);      // Function send the right file to
the client (if it exists)

String getContentType(String filename); // convert the file extension to
the MIME type

File fsUploadFile;                    // a File variable to temporarily
store the received file

// ----- Create class instances -----
// -----

ESP8266WebServer server(80);          // Create a webserver object that
listens for HTTP request on port 80

WebSocketsServer webSocket(81);       // create a websocket server on port 81

void setup() {

  Serial.begin(115200);                delay(1000);        Serial.print("\r\n");
  Serial.println("SmartBot v4.0");

  // ----- Setting GPIOs as Inputs/Outputs -----
  // -----

  pinMode(MRp, OUTPUT);  pinMode(MRm, OUTPUT);
  pinMode(MLp, OUTPUT);  pinMode(MLm, OUTPUT);
  pinMode(Button_pin, INPUT);
  pinMode(LDR_pin, INPUT);

  testSmartBot(); // Test SmartBot Movements
  startWiFi(); // Start Soft AP (Access Point)
  startSPIFFS(); // Start SPIFFS (SPI Flash Files System) and list all
contents
  startWebSocket(); // Start a WebSocket server

```

```

    startServer();          // Start a HTTP server with a file read handler and an
upload handler

    startmDNS();           // Start a mDNS responder

    startESPNow();        // Start ESP NOW communication

// ----- LDR values -----
LDR_value = analogRead(LDR_pin);    // Read the LDR's initial value
LDR_pr_value = LDR_value;           // Set them equal
}

void loop() {

    websocket.loop();                // constantly check for
websocket events

    server.handleClient();           // Listen for HTTP requests
from clients

    t = millis();                    // Read millis()

    if((t - last_10sec) > 100 * 1000) {    // Every 100sec check read
LDR value
        last_10sec = millis();
        checkLDR();                    // Check LDR value and act
    }

    if((t - last_03sec) > 0.3 * 1000) {    // Every 0.5 sec check read
LDR value
        last_03sec = millis();
        checkButton();                // Check Button
    }

    /*

    if((t - last_10sec) > 10 * 1000) {    // Every 10sec check read LDR
value
        counter++;
        bool ping = (counter % 2);
        int i = websocket.connectedClients(ping);
        USE_SERIAL.printf("%d Connected websocket clients ping: %d\n", i,
ping);
        last_10sec = millis();
    }

    */
}

```

```
// ----- Start Access Point -----  
-----  
  
void startWiFi(){  
    WiFi.softAP(ssid, password);           // Start the access point  
  
    Serial.print("\r\n    Access    Point    \");    Serial.print(ssid);  
    Serial.println("\n" started\r\n");  
  
    Serial.print("\n IP address:\t");    Serial.println(WiFi.softAPIP());  
    // Send the IP address of the ESP8266 to the computer  
  
}
```

## Project/mDNS\_dubot.ino

```
// Libraries to include
#include <ESP8266mDNS.h>           // Include the mDNS library
for ESP8266

// Constants & Variables
const char *mDNSname = "SmartBot"; // The name of the mDNS record
that will be broadcasted

// ----- Multicast Domain Name System (mDNS) -----
void startmDNS() {
    if (MDNS.begin(mDNSname)) {
        Serial.println("mDNS responder started");
    }
}
```

## Project/espnow\_esp8266\_receiver.ino

```
// Libraries to include

#include <espnow.h> //Important to use this instead of esp_now.h for esp8266
due to framework issues

// The following is the base data structure, must be the same on the Sender.
Used the base structure of a single command ["commandName",commandValue]

typedef struct {
    char commandName[32];
    int commandValue;
} CommandData;

CommandData espData; //Structure object

// Function to copy all recieved data to be able to process them

void onDataReceived(unit8_t * mac, unit8_t *incomingData, unit8_t len) {
    memcpy(&espData, incomingData, sizeof(espData));

    if (espData.commandName = "Command #1") { /* Code here to run when
"Command #1" is send */ }

    if (espData.commandName = "Command #2") { /* Code here to run when
"Command #2" is send */ }

    if (espData.commandName = "Command #3") { /* Code here to run when
"Command #3" is send */ }

    if (espData.commandName = "Command #4") { /* Code here to run when
"Command #4" is send */ }

    /* To Debug data transfer remove comments on the following lines
Serial.print("Bytes recieved: ");
Serial.println(len);
Serial.print("Command Name: ");
Serial.println(espData.commandName);
Serial.print("Command Value: ");
Serial.println(espData.commandValue);
Serial.println();
*/
```



```
}

// Function to initialize the ESP NOW protocol

void startESPNow() {
    if (esp_now_init() != 0) {
        Serial.println("Encountered error when starting ESP-NOW");
        return;
    }

    // Possibly add here code to register Peer if bidirectional
    communication

    esp_now_register_recv_cb(onDataReceived);
}
```

## Project/ESP32/espnow\_esp32\_sender.ino

```
// Libraries to include
#include <WiFi.h>
#include <esp_now.h>

// Constants & Variables
const char *ssid = "SmartBot"; // Match
the ssid to ESP8266

const char *password = "SmartBot"; // Match
the password to ESP8266

uint8_t broadcastAddress[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }; // To be
replaced with MAC Address of ESP8266

// The following is the base data structure, must be the same on the
Receiver. Used the base structure of a single command
["commandName", commandValue]

typedef struct {
    char commandName[32];
    int commandValue;
} CommandData;

CommandData espData; //Structure object
esp_now_peer_info_t peerInfo;

void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast Command Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Successful" :
"Failed");
}

void setup() {
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
}
```

```

if (esp_now_init() != ESP_OK) {
    Serial.println("Encountered error when starting ESP-NOW");
    return;
}

esp_now_register_send_cb(OnDataSent);

memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

if (esp_now_add_peer(&peerInfo) != ESP_OK) {
    Serial.println("Failed to add ESP8266");
    return;
}
}

void loop() {

    // Here goes the logic of what is to be send

    // Send the command packet on ESP NOW network
    // esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *)
    &espData, sizeof (espData));

    /* To Debug data transfer remove comments on the following lines
    if (result == ESP_OK) {
        Serial.println("Sent with success");
    }
    else {
        Serial.println("Error sending the data");
    }
    delay(2000);
    */
}

```

## Project/ESP32/espnow\_esp32\_sender\_ei.ino

```
// Libraries to include
#include <WiFi.h>
#include <esp_now.h>

// Constants & Variables
const char *ssid = "SmartBot"; // Match
the ssid to ESP8266

const char *password = "SmartBot"; // Match
the password to ESP8266

uint8_t broadcastAddress[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}; // To be
replaced with MAC Address of ESP8266

extern bool bb_found; // External declaration for bb_found

BoundingBox bb; // Global
declaration for bb

String cam_cmd; //
Declare a command variable for the camera label to use in this program

// The following is the base data structure, must be the same on the
Receiver. Used the base structure of a single command
["commandName",commandValue]

typedef struct {
    char commandName[32];
    int commandValue;
} CommandData;

CommandData espData; //Structure object
esp_now_peer_info_t peerInfo;

void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast Command Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Successful" :
"Failed");
}

void setup() {
    Serial.begin(115200);
```

```

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

if (esp_now_init() != ESP_OK) {
    Serial.println("Encountered error when starting ESP-NOW");
    return;
}

esp_now_register_send_cb(OnDataSent);

memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

if (esp_now_add_peer(&peerInfo) != ESP_OK) {
    Serial.println("Failed to add ESP8266");
    return;
}
}

void loop() {

    // Here goes the logic of what is to be send

    if (bb_found) {
        cam_cmd = bb.label;
    } else {
        cam_cmd = " ";
    }

    Serial.println("Command: ");
    Serial.println(cam_cmd);

    // Prepare the packet
    strcpy(espData.commandName, cam_cmd.c_str()); // Setting it up to
    send cam_cmd in the espData packet of ESP NOW
    espData.commandValue = 1; // 1 is used as an
    example in this case, holds no importance at this current moment

```

```
    // Send the command packet on ESP NOW network
    esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *)
&espData, sizeof (espData));

    /* To Debug data transfer remove comments on the following lines
if (result == ESP_OK) {
    Serial.println("Sent with success");
}
else {
    Serial.println("Error sending the data");
}
delay(2000);
*/
}
```

## Project/data/easy-en.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SmartBot v1</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link href="css/template.css?" rel="stylesheet" type="text/css">
  <script src="js/socket.js" type="text/javascript"></script>
</head>
<body>
  <div class="page-easy">
    <div class="top">
      <div class="left">
        <div class="buttons">
          <a class="tooltip" href="index.html">
          <span class="tooltiptext
bottom_right">Click here to return to the home page</span>
          </a>
          <a class="tooltip"
href="settings.html">
          <span class="tooltiptext
bottom_right">Click here to open settings</span>
          </a>
        </div>
      </div>
      <div class="center"><p style="text-align:
center">SmartBot</p></div>
      <div class="right" style="text-align:right;">
        <a id="myBtn" class="tooltip">
        <span class="tooltiptext
bottom_left">Click here for help</span>
        </a>
      </div>
    </div>
    <div id="myModal" class="modal">
```

```

        <div class="modal-content">
            <div class="modal-header">
                <span class="close">&times;</span>
                <h2>Help & Instructions </h2>
            </div>
            <div class="modal-body">
                <p>Press the keys to make the robot move.
    </p>

            </div>
            <div class="modal-footer">
                <h6>A. Hatzopoulos 2020</h6>
            </div>
        </div>
    </div>
    <script>
        var modal = document.getElementById("myModal");
        // Get the modal

        var btn = document.getElementById("myBtn");
        // Get the button that opens the modal

        var span = document.getElementsByClassName("close")[0];
        // Get the <span> element that closes the modal

        btn.onclick = function() {
            // When the user clicks on the button, open the
modal

            modal.style.display = "block"; }

        span.onclick = function() {
            // When the user clicks on <span> (x),
close the modal

            modal.style.display = "none"; }

        window.onclick = function(event) {
            // When the user clicks anywhere outside of the
modal, close it

            if (event.target == modal) {
                modal.style.display = "none";
            }
        }
    </script>
    <div class="main">
        <div class="main-intro">
            <table>
                <tr>

```



```

                <td>&nbsp;</td>
                <td></td>
                <td></td>
                <td></td>
                <td>&nbsp;</td>
            </tr>
            <tr>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td></td>
                <td></td>
                <td></td>
                <td>&nbsp;</td>
            </tr>
            <tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&
nbsp;</td></tr>
            <tr>
                <td>
                    <a class="tooltip">
                        
                        <span
class="tooltiptext forbutton">Drag the slider to increase or decrease the
speed</span>
                </td>
            </tr>

```

```

        </a>
    </td>
    <td colspan="3">
        <input class="enabled"
id="speed" type="range" min="0" max="1023" step="1" oninput="sendSpeed();"
value="512">
    </td>
    <td><p id="speed_value">512</p></td>
</tr>
</table>
</div>
</div>
</div>
</body>
</html>

```

## Project/data/medium-en.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SmartBot v1</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" type="text/css" href="lib/dragula.min.css">
  <link rel="stylesheet" type="text/css" href="css/template.css?">
  <script src="js/socket.js"></script>
</head>
<body>
  <div class="page-medium">
    <div class="top">
      <div class="left">
        <div class="buttons">
          <a class="tooltip" href="index.html">
          <span class="tooltiptext
bottom_right">Click here to return to the home page</span>
          </a>
          <a class="tooltip"
href="settings.html">
          <span class="tooltiptext
bottom_right">Click here to open settings</span>
          </a>
        </div>
      </div>
      <div class="center">
        <p>SmartBot</p>
        <a class="clear-action tooltip" href="">
          
          <span class="tooltiptext
bottom_right">Click here to delete the program.</span>
        </a>
        <a class="save-action tooltip" href="">
```

```

        
        <span class="tooltiptext
bottom_right">Click here to save the program to your device.</span>
    </a>
    <a class="load-action tooltip" href="">
        
        <span class="tooltiptext
bottom_right">Click here to load a program from your device.</span>
    </a>
    <div class="examples">
        <select class="choose-example">
            <option value="">Examples</option>
            <option
value="triangle">Triangle</option>
            <option value="circle">Tiny
Circle</option>
            <option
value="spiral">Spiral</option>
        </select>
    </div>
</div>
<div class="right" style="text-align:right;">
    <a id="myBtn" class="tooltip">
    <span class="tooltiptext
bottom_left">Click here for help</span>
    </a>
</div>
</div>
<div id="myModal" class="modal">
    <div class="modal-content">
        <div class="modal-header">
            <span class="close">&times;</span>
            <h2>Help & Instructions </h2>
        </div>
        <div class="modal-body">
            <p>Drag the Command Blocks and press the
Run button to make the robot perform a move. </p>
        </div>
    </div>

```

```

        <div class="modal-footer">
            <h6>A. Hatzopoulos 2020 - Thanks: Dethe
Elza for the Block Code Library</h6>
        </div>
    </div>
</div>
<script>
    var modal = document.getElementById("myModal");
    // Get the modal

    var btn = document.getElementById("myBtn");
    // Get the button that opens the modal

    var span = document.getElementsByClassName("close")[0];
    // Get the <span> element that closes the modal

    btn.onclick = function() {
        // When the user clicks on the button, open the
modal
        modal.style.display = "block"; }

    span.onclick = function() {
        // When the user clicks on <span> (x),
close the modal
        modal.style.display = "none"; }

    window.onclick = function(event) {
        // When the user clicks anywhere outside of the
modal, close it
        if (event.target == modal) {
            modal.style.display = "none";
        }
    }
</script>
    <div class="sendCommands">        <!--      Footer      and
Commands area is need only for testing purposes -->
        <div class="commands">
            <table>
                <tr>
                    <td
class="send_button"><p
id="snd_button" onclick="sendData();" oninput="sendData();">Send</p></td>
                    <td
class="type_cmd"><input
type="text" id="snd_data" class="rcv" placeholder="type your robot commands
and press Send"></td>
                    <td
class="send_button"><p
id="snd_button" onclick="sendData();" oninput="sendData();">Send</p></td>
                </tr>
            </table>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="main">
        <div class="script-column">
            <div class="script"> </div>
        </div>
    </div>

    <div class="bottom">
        <div class="menu-column">
            <div class="menu">
                <!--  -->
            </div>
        </div>
        <div class="canvas-column">
            <!-- The following is need, otherwise only Repeat Block is
shown -->
            <div class="canvas-placeholder"></div>
            <canvas class="canvas"></canvas>
        </div>

        </div> <!-- End of page -->
        <script src="lib/dragula.min.js"></script>
        <script src="js/util.js"></script>
        <script src="js/file.js"></script>
        <script src="js/blocks-medium.js"></script>
        <script src="js/menu-medium.js"></script>
        <script src="js/drag.js"></script>
        <script src="js/turtle-medium.js"></script>
        <script src="examples/spiral.js"></script>
        <script src="examples/tiny_circle.js"></script>
        <script src="examples/triangle.js"></script>

    </body>
</html>

```

## Project/data/hard-gr.html

```
<!DOCTYPE html>
<html lang="el">
<head>
  <title>SmartBot v1</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <!--Include Basic SmartBot CSS & functionality -->
  <link rel="stylesheet" type="text/css" href="lib/dragula.min.css">
  <link rel="stylesheet" type="text/css" href="css/template.css?">
  <script src="js/socket.js"></script>
  <!-- Include CodeMirror CSS -->
  <link rel="stylesheet" href="cm/codemirror.min.css">
  <!-- CodeMirror Styling Code -->
  <link rel="stylesheet" href="cm/dracula.min.css">
  <!-- Theme for CodeMirror Editor -->
  <script src="cm/codemirror.min.js"></script>
  <!-- CodeMirror functionality -->
  <script src="cm/clike.min.js"></script>
  <!-- C++ highlight functionality -->
  <script src="cm/python.min.js"></script>
  <!-- Python highlight functionality -->
</head>
<body>
  <div class="page-hard">
    <div class="top">
      <div class="left">
        <div class="buttons">
          <a class="tooltip" href="index.html">
          <span class="tooltiptext
bottom_right">Πάτησε εδώ για επιστροφή στην αρχική οθόνη.</span>
          </a>
          <a class="tooltip"
href="settings.html">
          <span class="tooltiptext
bottom_right">Πάτησε εδώ για τις ρυθμίσεις.</span>
          </a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

</div>
<div class="center">
    <p>SmartBot</p>
</div>
<div class="right" style="text-align:right;">
    <a id="myBtn" class="tooltip">
    <span class="tooltiptext
bottom_left">Πάτησε εδώ για βοήθεια.</span>
    </a>
</div>
</div>
<div id="myModal" class="modal">
    <div class="modal-content">
        <div class="modal-header">
            <span class="close">&times;</span>
            <h2>Βοήθεια & Οδηγίες Χρήσης </h2>
        </div>
        <div class="modal-body">
            <p>Γράψε τον κώδικα κίνησης και πάτα το εικονίδιο
εκτέλεσης για να προσομοιωθεί το πρόγραμμα.</p>
            <p>Πάτα το πλήκτρο Run για να εκτελέσει το
πρόγραμμα το ρομπότ.</p>
        </div>
        <div class="modal-footer">
            <h6>A. Χατζόπουλος 2024</h6>
        </div>
    </div>
</div>
<script>
    var modal = document.getElementById("myModal");
    // Get the modal

    var btn = document.getElementById("myBtn");
    // Get the button that opens the modal

    var span = document.getElementsByClassName("close")[0];
    // Get the <span> element that closes the modal

    btn.onclick = function() {
        // When the user clicks on the button, open the modal
        modal.style.display = "block"; }

```



```

        span.onclick = function() {
            // When the user clicks on <span> (x), close the
modal
            modal.style.display = "none"; }
        window.onclick = function(event) {
            // When the user clicks anywhere outside of the modal,
close it
            if (event.target == modal) {
                modal.style.display = "none";
            }
        }
    </script>
    <div class="main">
        <div class="leftcolumn">
            <p>Επεξεργαστής Κώδικα</p>
            <textarea id="code" name="code">Speed(512)
Forward(100)
Right90(1)
Forward(50)</textarea>
            <div class="buttonline">
                <a class="save-h-action tooltip" href="">
                    
                    <span class="tooltiptext
bottom_right">Πάτησε εδώ για να αποθηκεύσεις το πρόγραμμα στη συσκευή
σου.</span>
                </a>
                <a class="load-h-action tooltip" href="">
                    
                    <span class="tooltiptext
bottom_right">Πάτησε εδώ για να φορτώσεις ένα πρόγραμμα από τη συσκευή
σου.</span>
                </a>
                <input type="file" id="fileInput"
style="display: none;">
                <button class="wrap-action tooltip button
off" data-wrap="false">Wrap
                    <span class="tooltiptext
bottom_right">Πάτησε εδώ για να αναδιπλώσεις τον κώδικα.</span>
                </button>
                <button class="animate-action tooltip
button" data-status="on">Animation

```

```

        <span class="tooltiptext
bottom_right">Πάτησε εδώ για τον έλεγχο κινούμενου γραφικού.</span>
        </button>
    </div>
</div>
<script>
// Initialize CodeMirror with Python mode by default
var editor =
CodeMirror.fromTextArea(document.getElementById("code"), {
    lineNumbers: true,
    lineWrapping: false,
    mode: "python", // Default to Python mode //
"text/x-c++src", Default to C++ mode
    theme: "dracula"
});
</script>
<div class="midcolumn">
    <div>
        <p>Προσομοιωτής</p>
        <div id="canvas" style="width: 400px;
height: 400px"></div>
        <pre id="output" style="display:
none;"></pre>
    </div>
    <div class="canvabuttonline">
        <a class="canvarun-action tooltip"
href="">
            
            <span class="tooltiptext
bottom_right">Πάτησε εδώ για εκτέλεση της προσομοίωσης.</span>
        </a>
        <a class="saveimg-action tooltip" href="">
            
            <span class="tooltiptext
bottom_right">Πάτησε εδώ για να αποθηκεύσεις την προσομοίωση στη συσκευή
σου.</span>
        </a>
        <a class="canvaclear-action tooltip"
href="">
            

```

```

        <span class="tooltiptext
bottom_right">Πάτησε εδώ για καθαρισμό της προσομοίωσης.</span>
        </a>
    </div>
    <br>
    <div class="commandContainer">
        <label for="commandInput">Τεραμτικό:
    </label>
        <input type="text" id="commandInput"
placeholder="Γράψε εδώ...">
        <button class="cmdrun-action tooltip
button">Run
        <span class="tooltiptext
bottom_right">Πάτησε εδώ για την εκτέλεση εντολής.</span>
    </button>
    </div>
</div>
<div class="rightcolumn">
    <p>Λεξικό εντολών</p>
    <div class="container">
        <code class="code-reference">
            <div class="row header">
                <div
class="command">Εντολή</div>
                <div class="description"
style="padding: 0; margin: 0 0 0 5px;">Περιγραφή</div>
            </div>
            <div class="scrollable">
                <div class="row">
                    <div
class="command">Forward(distance)</div>
                    <div
class="description">
                        Μετακινήστε το
                        αυτοκίνητο προς τα εμπρός κατά την καθορισμένη απόσταση.
                    </div>
                </div>
                <div class="example">
                    π.χ.<br>
                    Forward(100)
                </div>
            </div>
        </code>
    </div>
</div>

```

```

</div>
<div class="row">
  <div
class="command">ForwardLeft (distance)</div>
  <div
class="description">
    Μετακινήστε το
    αυτοκίνητο προς τα εμπρός με γωνία 45 προς τα αριστερά κατά την καθορισμένη
    απόσταση.
  <div
class="example">
    π.χ.<br>
    ForwardLeft (100)
  </div>
</div>
</div>
<div class="row">
  <div
class="command">ForwardRight (distance)</div>
  <div
class="description">
    Μετακινήστε το
    αυτοκίνητο προς τα εμπρός με γωνία 45 προς τα δεξιά κατά την καθορισμένη
    απόσταση.
  <div
class="example">
    π.χ.<br>
    ForwardRight (100)
  </div>
</div>
</div>
<div class="row">
  <div
class="command">RotateCCW (angle)</div>
  <div
class="description">
    Περιστρέψτε το
    αυτοκίνητο αριστερόστροφα κατά την καθορισμένη γωνία.
  <div
class="example">
    π.χ.<br>

```

<pre> RotateCCW(45) </pre>	<pre> &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div       π.χ.&lt;br&gt;       Left90(1)     &lt;/div&gt;   &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div       π.χ.&lt;br&gt; </pre>
<pre> class="command"&gt;Left90(1)&lt;/div&gt; class="description"&gt; αυτοκίνητο αριστερά κατά 90 μοίρες. class="example"&gt; </pre>	<pre> &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div       π.χ.&lt;br&gt;       Left90(1)     &lt;/div&gt;   &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div       π.χ.&lt;br&gt; </pre>
<pre> class="command"&gt;RotateCW(angle)&lt;/div&gt; class="description"&gt; αυτοκίνητο δεξιόστροφα κατά την καθορισμένη γωνία. class="example"&gt; </pre>	<pre> &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div       π.χ.&lt;br&gt; </pre>
<pre> RotateCW(45) </pre>	<pre> &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div </pre>
<pre> class="command"&gt;Right90(1)&lt;/div&gt; class="description"&gt; αυτοκίνητο δεξιά κατά 90 μοίρες. class="example"&gt; </pre>	<pre> &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div class="row"&gt;   &lt;div     &lt;div       Περιστρέψτε το     &lt;div </pre>

π.χ.<br>

Right90 (1)

</div>

</div>

</div>

<div class="row">

<div

class="command">Backward (distance)</div>

<div

class="description">

αυτοκίνητο προς τα πίσω κατά την καθορισμένη απόσταση.

Μετακινήστε το

<div

class="example">

π.χ.<br>

Backward (100)

</div>

</div>

</div>

<div class="row">

<div

class="command">BackwardLeft (distance)</div>

<div

class="description">

αυτοκίνητο προς τα πίσω με γωνία 45 προς τα δεξιά κατά την καθορισμένη απόσταση.

Μετακινήστε το

<div

class="example">

π.χ.<br>

BackwardLeft (100)

</div>

</div>

</div>

<div class="row">

<div

class="command">BackwardRight (distance)</div>

<div

class="description">

Μετακινήστε το αυτοκίνητο προς τα πίσω με γωνία 45 προς τα αριστερά κατά την καθορισμένη απόσταση.

```
class="example">
    <div
        π.χ.<br>
        BackwardRight(100)
    </div>
</div>
</div>
```

```
<div class="row">
    <div
class="command">Speed(speed)</div>
    <div
class="description">
```

Ρυθμίστε την ταχύτητα του αυτοκινήτου στην καθορισμένη τιμή, περιορισμένη σε (1 - 1023).

```
<div
class="example">
    π.χ.<br>
    Speed(512)
</div>
</div>
</div>
```

```
<div class="row">
    <div
class="command">Repeat(number){command block}</div>
    <div
class="description">
```

Επαναλάβετε μια σειρά εντολών για έναν καθορισμένο αριθμό φορών.

```
<div
class="example">
    π.χ.<br>
    Repeat(4)
    {<br>
        Forward(100)<br>
        Right90(1)<br>
```

```
    }
</div>
```

```

        </div>
    </div>
    <div class="row">
        <div
class="command">PathPlan</div>
        <div
class="description">
            Αυτή η εντολή
            είναι μόνο για το Τερματικό, όχι για τον κώδικα<br>
            Αποστέλει τον
            κώδικα του χρήστη στο ρομπότ.
        </div>
class="example">
            Τερματικό: PathPlan
        </div>
    </div>
</div>
</code>
</div>
</div>
</div>
</div>
</div> <!-- End of page -->
<script src="lib/dragula.min.js"></script>
<script src="cm/jquery.min.js"></script>
<script src="sk/skulpt.min.js"></script>
<script src="sk/skulpt-stdlib.js"></script>
<script src="js/custom-cmd.js"></script>
<script> <!-- jQuery script for button functionalities -->
    (function($) {
        function findTurtleCanvas() {
            var canvas = $('#canvas').find('canvas');
            if (canvas.length && canvas.length > 1)
                return canvas.get(1);
            else
                return false;
        }
        function clearCanvas() {

```



```

"turtle.Turtle()\n";
    var resetScript = "import turtle\n" +
    Sk.misceval.asyncToPromise(function() {
        return
Sk.importMainWithBody("<preset_script>", false, resetScript, true);
    }).then(function(mod) {
        console.log('Canvas          cleared
successfully');
    }, function(err) {
        console.log('Error when clearing Canvas:',
err.toString());
    });
}
$('#fileInput').change(function(event) {
    var file = event.target.files[0];
    var reader = new FileReader();
    reader.onload = function(e) {
        var fileContent = e.target.result;
        editor.setValue(fileContent);
    };
    reader.readAsText(file);
});
$('.wrap-action').click(function() {
    var wrapping = $(this).data('wrap') ||
$(this).data('wrap') == 'true';
    $(this).data('wrap', !wrapping);
    editor.setOption("lineWrapping", !wrapping);
    if (wrapping) {
        $(this).addClass('off');
    } else {
        $(this).removeClass('off');
    }
});
$('.animate-action').click(function() {
    if($(this).data('status') == 'on')
    {
        $(this).data('status', 'off');
        $(this).addClass('off');
    }
}

```

```

        else
        {
            $(this).data('status','on');
            $(this).removeClass('off');
        }
    });

    $('.saveimg-action').click(function(e) {
        var filename = $('input[name="filename"]').val()
        ? $('input[name="filename"]').val().replace('.py','.png') : 'PathPlan-
        canvas.png';

        var canvas = findTurtleCanvas();
        var img = canvas.toDataURL('image/png');
        $(this).attr({
            'download': filename,
            'href': img
        });
    });

    $('.canvaclear-action').click(function() {
        event.preventDefault();
        clearCanvas();
    });

    $('.canvarun-action').click(function() {
        event.preventDefault();
        runCode(); //Run the code from editor on Canva
    });

    $('.cmdrun-action').click(function() {
        cmdRun(); //Run the code from editor on ESP
    });

    $('.save-h-action').click(function() {
        var code = editor.getValue();
        var blob = new Blob([code], {type:
'text/plain'});

        var url = URL.createObjectURL(blob);
        $(this).attr('href', url);
        $(this).attr('download', 'code.txt');
    });

    $('.load-h-action').click(function(event) {
        event.preventDefault();

```

```
        $('#fileInput').click();
    });
})(jQuery);
</script>
</body>
</html>
```

## Project/data/hard-en.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SmartBot v1</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <!--Include Basic SmartBot CSS & functionality -->
  <link rel="stylesheet" type="text/css" href="lib/dragula.min.css">
  <link rel="stylesheet" type="text/css" href="css/template.css?">
  <script src="js/socket.js"></script>
  <!-- Include CodeMirror CSS -->
  <link rel="stylesheet" href="cm/codemirror.min.css">
  <!-- CodeMirror Styling Code -->
  <link rel="stylesheet" href="cm/dracula.min.css">
  <!-- Theme for CodeMirror Editor -->
  <script src="cm/codemirror.min.js"></script>
  <!-- CodeMirror functionality -->
  <script src="cm/clike.min.js"></script>
  <!-- C++ highlight functionality -->
  <script src="cm/python.min.js"></script>
  <!-- Python highlight functionality -->
</head>
<body>
  <div class="page-hard">
    <div class="top">
      <div class="left">
        <div class="buttons">
          <a class="tooltip" href="index.html">
          <span class="tooltiptext
bottom_right">Click here to return to the home screen.</span>
          </a>
          <a class="tooltip"
href="settings.html">
          <span class="tooltiptext
bottom_right">Click here for settings.</span>
          </a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        </div>
        <div class="center">
            <p>SmartBot</p>
        </div>
        <div class="right" style="text-align:right;">
            <a id="myBtn" class="tooltip">
                <span class="tooltiptext
bottom_left">Click here for help.</span>
            </a>
        </div>
    </div>
    <div id="myModal" class="modal">
        <div class="modal-content">
            <div class="modal-header">
                <span class="close">&times;</span>
                <h2>Help & Instructions </h2>
            </div>
            <div class="modal-body">
                <p>Write the animation code and press the run
icon to simulate the program.</p>
                <p>Press the Run button to make the robot run the
program.</p>
            </div>
            <div class="modal-footer">
                <h6>A. Hatzopoulos 2024</h6>
            </div>
        </div>
    </div>
    <script>
        var modal = document.getElementById("myModal");
        // Get the modal

        var btn = document.getElementById("myBtn");
        // Get the button that opens the modal

        var span = document.getElementsByClassName("close")[0];
        // Get the <span> element that closes the modal

        btn.onclick = function() {
            // When the user clicks on the button, open the modal

            modal.style.display = "block"; }
    </script>

```

```

        span.onclick = function() {
            // When the user clicks on <span> (x), close the
modal
            modal.style.display = "none"; }
        window.onclick = function(event) {
            // When the user clicks anywhere outside of the modal,
close it
            if (event.target == modal) {
                modal.style.display = "none";
            }
        }
    </script>
    <div class="main">
        <div class="leftcolumn">
            <p>Code Editor</p>
            <textarea id="code" name="code">Speed(512)
Forward(100)
Right90(1)
Forward(50)</textarea>
            <div class="buttonline">
                <a class="save-h-action tooltip" href="">
                    
                    <span class="tooltiptext
bottom_right">Click here to save the program to your device.</span>
                </a>
                <a class="load-h-action tooltip" href="">
                    
                    <span class="tooltiptext
bottom_right">Click here to save the program to your device.</span>
                </a>
                <input type="file" id="fileInput"
style="display: none;">
                <button class="wrap-action tooltip button
off" data-wrap="false">Wrap
                    <span class="tooltiptext
bottom_right">Click here to toggle the wrap of the code.</span>
                </button>
                <button class="animate-action tooltip
button" data-status="on">Animation
                    <span class="tooltiptext
bottom_right">Click here to toggle the animation.</span>

```

```

        </button>
    </div>
</div>
<script>
    // Initialize CodeMirror with Python mode by default
    var editor =
CodeMirror.fromTextArea(document.getElementById("code"), {
        lineNumbers: true,
        lineWrapping: false,
        mode: "python", // Default to Python mode //
"text/x-c++src", Default to C++ mode
        theme: "dracula"
    });
</script>
<div class="midcolumn">
    <div>
        <p>Simulator Canvas</p>
        <div id="canvas" style="width: 400px;
height: 400px"></div>
        <pre id="output" style="display:
none;"></pre>
    </div>
    <div class="canvabuttonline">
        <a class="canvarun-action tooltip"
href="">
            
            <span class="tooltiptext
bottom_right">Click here to run the simulation.</span>
        </a>
        <a class="saveimg-action tooltip" href="">
            
            <span class="tooltiptext
bottom_right">Click here to save the simulation to your device.</span>
        </a>
        <a class="canvaclear-action tooltip"
href="">
            
            <span class="tooltiptext
bottom_right">Click here to clear the simulation.</span>

```

```

        </a>
    </div>
    <br>
    <div class="commandContainer">
        <label          for="commandInput">Terminal:
</label>
        <input      type="text"      id="commandInput"
placeholder="Write here...">
        <button      class="cmdrun-action      tooltip
button">Run
        <span          class="tooltiptext
bottom_right">Click here to run the terminal command.</span>
        </button>
    </div>
</div>
<div class="rightcolumn">
    <p>Code References</p>
    <div class="container">
        <code class="code-reference">
            <div class="row header">
                <div
class="command">Command</div>
                <div      class="description"
style="padding: 0; margin: 0 0 0 5px;">Description</div>
            </div>
            <div class="scrollable">
                <div class="row">
                    <div
class="command">Forward(distance)</div>
                    <div
class="description">
                        Move the car
                        forward by the specified distance.
                    </div>
                </div>
            </div>
            <div class="example">
                Example:<br>
                Forward(100)
            </div>
        </code>
    </div>
</div>
</div>

```



```

class="command">ForwardLeft (distance)</div>
class="description">
forward with a 45 angle to the left by the specified distance.
class="example">
    Example:<br>
    ForwardLeft (100)
class="command">ForwardRight (distance)</div>
class="description">
forward with a 45 angle to the right by the specified distance.
class="example">
    Example:<br>
    ForwardRight (100)
class="command">RotateCCW (angle)</div>
class="description">
counter-clockwise by the specified angle.
class="example">
    Example:<br>
    RotateCCW (45)

```

```

<div class="row">
  <div
  <div
    Move the car
  <div
  </div>
  </div>
</div>
<div class="row">
  <div
  <div
    Move the car
  <div
  </div>
  </div>
</div>
<div class="row">
  <div
  <div
    Rotate the car
  <div

```

```
class="command">Left90(1)</div>
```

```
class="description">  
left by 90 degrees.
```

```
class="example">  
Example:<br>
```

```
class="command">RotateCW(angle)</div>
```

```
class="description">  
clockwise by the specified angle.
```

```
class="example">  
Example:<br>
```

```
RotateCW(45)
```

```
class="command">Right90(1)</div>
```

```
class="description">  
right by 90 degrees.
```

```
class="example">
```

```
</div>  
</div>  
</div>  
<div class="row">  
  <div  
  <div  
    Rotate the car  
  <div  
    Left90(1)  
  </div>  
</div>  
</div>  
<div class="row">  
  <div  
  <div  
    Rotate the car  
  <div  
    RotateCW(45)  
  </div>  
</div>  
</div>  
<div class="row">  
  <div  
  <div  
    Rotate the car  
  <div
```

```

Example:<br>

Right90 (1)

class="command">Backward (distance)</div>
class="description">
backward by the specified distance.
class="example">
Example:<br>
Backward (100)

class="command">BackwardLeft (distance)</div>
class="description">
backward with a 45 angle to the right by the specified distance.
class="example">
Example:<br>
BackwardLeft (100)

class="command">BackwardRight (distance)</div>
class="description">

```

```

</div>
</div>
</div>
<div class="row">
<div
<div
Move the car
<div
</div>
</div>
</div>
<div class="row">
<div
<div
Move the car
<div
</div>
</div>
</div>
<div class="row">
<div
<div

```

```

Move the car
backward with a 45 angle to the left by the specified distance.
class="example">
    Example:<br>
    BackwardRight(100)
</div>
</div>
</div>
<div class="row">
class="command">Speed(speed)</div>
class="description">
    Set the speed of
the car to the specified value, limited to (1 - 1023).
class="example">
    Example:<br>
    Speed(512)
</div>
</div>
</div>
<div class="row">
class="command">Repeat(number){command block}</div>
class="description">
    Repeat a
sequence of commands for a specified number of times.
class="example">
    Example:<br>
    {<br>
    Forward(100)<br>
    Right90(1)<br>
    }

```

```

        </div>
    </div>
</div>
<div class="row">
    <div
class="command">PathPlan</div>
    <div
class="description">
        This command is
only for the Terminal, not the actual code<br>
        It sends the
user's code to the car.
    <div
class="example">
        Terminal:
PathPlan
    </div>
    </div>
</div>
</div>
</code>
</div>
</div>
</div>
</div> <!-- End of page -->
<script src="lib/dragula.min.js"></script>
<script src="cm/jquery.min.js"></script>
<script src="sk/skulpt.min.js"></script>
<script src="sk/skulpt-stdlib.js"></script>
<script src="js/custom-cmd.js"></script>
<script> <!-- jQuery script for button functionalities -->
    (function($) {
        function findTurtleCanvas() {
            var canvas = $('#canvas').find('canvas');
            if (canvas.length && canvas.length > 1)
                return canvas.get(1);
            else
                return false;
        }
    })

```

```

        function clearCanvas() {
            var resetScript = "import turtle\n" +
"turtle.Turtle()\n";
            Sk.misceval.asyncToPromise(function() {
                return
Sk.importMainWithBody("<preset_script>", false, resetScript, true);
            }).then(function(mod) {
                console.log('Canvas          cleared
successfully');
            }, function(err) {
                console.log('Error when clearing Canvas:',
err.toString());
            });
        }
        $('#fileInput').change(function(event) {
            var file = event.target.files[0];
            var reader = new FileReader();
            reader.onload = function(e) {
                var fileContent = e.target.result;
                editor.setValue(fileContent);
            };
            reader.readAsText(file);
        });
        $('.wrap-action').click(function() {
            var wrapping = $(this).data('wrap') ||
$(this).data('wrap') == 'true';
            $(this).data('wrap', !wrapping);
            editor.setOption("lineWrapping", !wrapping);
            if (wrapping) {
                $(this).addClass('off');
            } else {
                $(this).removeClass('off');
            }
        });
        $('.animate-action').click(function() {
            if($(this).data('status') == 'on')
            {
                $(this).data('status', 'off');
                $(this).addClass('off');
            }
        });
    });

```

```

        }
        else
        {
            $(this).data('status','on');
            $(this).removeClass('off');
        }
    });
    $('.saveimg-action').click(function(e) {
        var filename = $('input[name="filename"]').val()
?   $('input[name="filename"]').val().replace('.py','.png')    :   'PathPlan-
canvas.png';

        var canvas = findTurtleCanvas();
        var img = canvas.toDataURL('image/png');
        $(this).attr({
            'download': filename,
            'href': img
        });
    });
    $('.canvaclear-action').click(function() {
        event.preventDefault();
        clearCanvas();
    });
    $('.canvarun-action').click(function() {
        event.preventDefault();
        runCode(); //Run the code from editor on Canva
    });
    $('.cmdrun-action').click(function() {
        cmdRun(); //Run the code from editor on ESP
    });
    $('.save-h-action').click(function() {
        var code = editor.getValue();
        var blob = new Blob([code], {type:
'text/plain'});

        var url = URL.createObjectURL(blob);
        $(this).attr('href', url);
        $(this).attr('download', 'code.txt');
    });
    $('.load-h-action').click(function(event) {

```

```
        event.preventDefault();
        $('#fileInput').click();
    });
})(jQuery);
</script>
</body>
</html>
```



## Project/data/css/template.css

```
.clearfix:after { content: "."; display: block; clear: both;
visibility: hidden; line-height: 0; height: 0;}

.clearfix { display: inline-block;}

html[xmlns] .clearfix { display: block;}

* html .clearfix { height: 1%;}

/* ----- All Pages ----- */

html, body{ width:100%; height: 100%; padding: 0; border: 0; margin: 0;}

body {background-color:#333; font-family: Verdana, Arial, Helvetica, sans-
serif; }

h1, h2, h3, h4, h5, h6, p {margin:0; padding: 5px; color:#fff; font-weight:
normal; text-align: center; margin: 0em; }

h1 {font-size: 30px;} h2 {font-size: 26px;}

h3 {font-size: 22px;} h6 {font-size: 12px;}

p {font-size: 18px;}

* { box-sizing: border-box;}

/* ----- All Pages Icons ----- */

img.el, img.en {max-width: 40px; height: auto;}

img.el {margin-right:20px;}

img.help, img.icon { max-height: 30px; width: auto; cursor:pointer; }

img.help { margin-right: 10px; }

img.icon { margin-right: 20px; }

/* ----- All Pages Animations ----- */

img.anim:hover { animation-name: scale-up-center; animation-duration:
1.3s; animation-timing-function: ease;

animation-delay: 0s; animation-iteration-count: 1; animation-direction:
normal; animation-fill-mode: none;}

/* ----- Intro page -----*/

.page-index { width:100%; height:100%; display:table;}

.page-index .top { margin: 0; padding: 0; display: table-row; vertical-
align: top; height: 80px;}

.page-index .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-index .top .left .flags {padding-left:10px;}
```

```

.page-index .top .right {float:left; width:20%; padding: 25px 50px 0 0;}
.page-index .top .center {float:left; width:60%; text-align: center;
padding: 25px 0 0 0;}

.page-index .main {display: table; width: 100%; height: 100%;}

.page-index .main-intro {display: table-cell; vertical-align: middle;}

.page-index .main .onethird {float:left; width:33.3333%; text-align:
center;}

.page-index p.top-msg {margin: 0 auto 50px auto; text-align: center; font-
size: 22px;}

.page-index img.easy, .page-index img.medium, .page-index img.hard {max-
width: 250px; margin:0 auto; width: 100%; height: 100%;}

.page-index a.level {padding:15px; text-align:center; margin: 0 auto;}

/* ----- Easy page -----
- */

.page-easy { width:100%; height:100%; display:table; background-color:
#333;}

.page-easy .top { margin: 0; padding: 0; display: table-row; vertical-align:
top; height: 80px; background-color: #4f4f4f;}

.page-easy .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-easy .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-easy .top .center {float:left; width:60%; text-align: center; padding:
25px 0 0 0;}

.page-easy .main {display: table; width: 100%; height: 100%; text-align:
center; }

.page-easy .main-intro {display: table-cell; vertical-align: middle;}

.page-easy .main .main-intro table {border: 1px solid #4f4f4f; background-
color:#4f4f4f; padding: 30px 25px 20px 25px;
border-radius: 10px; margin:20px auto 0 auto;}

.page-easy .main .main-intro td {padding: 5px;}

.page-easy .main .main-intro p#speed_value {background-color:#fff;
color:#333; font-size: 20px; text-align: center; padding: 2px 5px; max-
width: 100px; border-radius: 5px; height: 2.5em; line-height: 2.4em;
margin: 5px 0 0 0;}

/* ---- Easy page Speed's Slider ---- */

.page-easy input[type=range] { -webkit-appearance: none; outline: none;
width: 100%; }

.page-easy input[type=range]::-webkit-slider-thumb { -webkit-appearance:
none; border: 0px; height: 20px; width: 20px; margin-top: -8px;
border-radius: 10px; } /*
slider's knob*/

.page-easy input.enabled[type=range]::-webkit-slider-thumb { background:
red; cursor: pointer; box-shadow: 1px 1px 2px #777, 0px 0px 1px #777; }

```

```

.page-easy input[type=range]::-webkit-slider-runnable-track { width:
inherit; height: 5px; background: #fde900; border-radius: 1px; }

.page-easy input.enabled[type=range]::-webkit-slider-runnable-track {
cursor: pointer; }

/* ---- Easy page Blocks ---- */

.page-easy img.block-icon { max-width: 100px; width: auto; cursor:pointer;
border-radius: 10px; transition: 0.3s; display: inline-block;}

.page-easy img.block-icon:hover {opacity: 0.5;}

/* ----- Medium page -----
--- */

.page-medium {width:100%; height:100%; display:table; background-color:
#333;}

.page-medium .top { margin: 0; padding: 0; display: table-row; vertical-
align: top; height: 80px; background-color: #4f4f4f;}

.page-medium .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-medium .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-medium .top .center {float:left; width:60%; text-align: center;
padding: 25px 0 0 0;}

.page-medium .top p { text-align: center; display: inline-block; margin-
right:100px; font-size:26px; }

.page-medium .top a { display: inline-block; }

.page-medium .top .examples {display:inline-block; padding:0; vertical-
align: top; }

.page-medium .top select.choose-example { height: 37px; border-radius: 7px;
margin: 0; padding: 0 10px; }

.page-medium .main {width: 100%; height:100%; display: table; vertical-
align: middle; margin: 0; padding:0; }

.page-medium .main .script-column {display: table-cell; vertical-align:
middle;}

.page-medium .main .script{ width: 100vw; height: 120px; white-space:
nowrap; overflow-x: scroll; /* border: 1px dotted red; */ }

.page-medium .main .script.over{ background-color: CadetBlue; }

.page-medium .bottom {width:100%; height: 100px; display: table-row;
vertical-align: bottom; margin: 0; padding: 10px; background-color:
#4f4f4f;}

.page-medium .bottom .menu-column {width: 100vw; white-space: nowrap;
overflow-x: scroll;}

.page-medium .bottom .menu{ border-radius: 7px; padding: 20px 20px; /*
overflow-y: auto; overflow-x: scroll; */ }

.page-medium .bottom .menu.over{ background-color: BurlyWood; }

```

```

.page-medium .canvas-column {display:none; border:1px blue solid;}
    /* Without canvas in HTML Blocks are not working */

/* sendCommands area only for testing purposes - Must be deleted */

.page-medium .sendCommands {height: 100px; display: table-row; vertical-align: bottom; margin: 0; padding: 10px; }

.page-medium .sendCommands .commands {}

.page-medium .sendCommands .commands table {width: 100vw; margin:0; padding:0;}

.page-medium .sendCommands .commands table td.send_button {width: 100px; margin:0; padding:0;}

.page-medium .sendCommands .commands table td.type_cmd { margin:0; padding:0;}

.page-medium .sendCommands .commands p {width:100px; background-color: red; border-radius:5px; cursor: pointer;}

.page-medium .sendCommands .commands table td.type_cmd input#snd_data { width:100%; margin:0; padding: 6px; border-radius: 4px;}

/* sendCommands area only for testing purposes - Must be deleted */

/* ---- Medium page Blocks - Script area Blocks ---- */

.page-medium .main .script .block { /* float:left; */ display: inline-block; margin: 0 1px 0 0; height: 80px; padding: 0;} /* float is not working with overflow */

.page-medium .main .script .block img {width: 80px; height:80px; border-radius:5px;}

.page-medium .main .script img.block-icon { /* float: left; */ margin: 0 1px 0 0; max-width: 80px; width: auto; cursor: pointer; border-radius: 10px; transition: 0.3s; display: inline-block ;}

.page-medium .main .script img.block-icon.run-action { vertical-align: bottom; margin: 0; padding: 0; float:left;}

.page-medium .main .script img.block-icon.run-action:hover {opacity: 0.5;}

/* ---- Medium page Blocks - Menu area Blocks ---- */

.page-medium .bottom .menu .block { /* float:left; */ display: inline-block; margin: 0 20px 0 0px; height: 80px; padding: 0; }

.page-medium .bottom .menu .block img {width: 80px; height:80px; border-radius:5px;}

```

```

.page-medium .block{ cursor: move; }

.page-medium .block.running{ outline: 5px solid red; }
/* Do I need this? */

.page-medium .block.next{ outline: 5px solid green; } /*
Do I need this? */

.page-medium .block.dragging{ opacity: 0.4; }

.page-medium .block input{ width: 5em; margin: -3px 0 0 0; display: block;
width: 80px; text-align: center; padding: 0;} /* Dethe */

/* ----- Dethe Code Block. css -----
container */

.container{ margin: 0 -6px 0 20px; padding-bottom: 10px; border: 1px
inset #021429; border-radius: 7px; border-right: 0; background-color:
LightSteelBlue; border-radius: 7px 0 0 7px;}

.over .container{ background-color: CadetBlue; }

/* ----- Dethe Code Block. css -----

.canvas-column{ width: 50%; height: 85%; float: right; margin-right: 10px; }

.canvas-placeholder{ width: 100%; height: 85%; }

.canvas{ background-color: white; position: absolute; }

/* ----- Dethe Code Block. css -----
*/

/* ----- Hard page -----
- */

.page-hard {width:100%; height:100%; display:table; background-color: #333;}

.page-hard .top { margin: 0; padding: 0; display: table-row; vertical-align:
top; height: 80px; background-color: #4f4f4f;}

.page-hard .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-hard .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-hard .top .center {float:left; width:60%; text-align: center; padding:
25px 0 0 0;}

.page-hard .top p { text-align: center; display: inline-block; font-
size:26px;}

.page-hard .top a { display: inline-block;}

```

```

.page-hard .main { margin: 0; padding: 0 20px; display: flex; justify-
content: space-between; align-items: flex-start;}

.page-hard .main p {padding: 10px 0;}

.page-hard .main img.icon {margin: 0 5px;}

.page-hard .main .button {background-color: white; color: black; border: 2px
solid black; padding: 5px 10px; text-align: center; text-decoration: none;
display: inline-block; font-size: 14px; cursor: pointer; border-radius:
8px;}

.page-hard .main .button.off {background-color: #4f4f4f; color: #aaa;}

.page-hard .main .leftcolumn {min-width: 300px; width: 30%; box-sizing:
border-box; flex: 0 0 30%;}

.page-hard .main .leftcolumn .CodeMirror {height: 450px;}

.page-hard .main .leftcolumn .CodeMirror-scroll {padding-bottom: 20px;}

.page-hard .main .leftcolumn .buttonline {display: flex; align-items:
center; padding: 0 10px; margin-top: 10px;}

.page-hard .main .leftcolumn .buttonline button {margin-left: 5px; min-
width: min-content;}

.page-hard .main .leftcolumn .buttonline button:last-child {margin-left:
auto;}

.page-hard .main .midcolumn {min-width: 400px; width: 30%; box-sizing:
border-box; margin-left: 5%; margin-right: 5%; flex: 0 0 30%; text-align:
center;}

.page-hard .main .midcolumn .commandContainer {margin: 0 auto; width:400px;
align-items: center; justify-content: space-between; display: flex;}

.page-hard .main .midcolumn .commandContainer label {font-size: 16px; color:
white;}

.page-hard .main .midcolumn .commandContainer input {flex-grow: 1; margin: 0
10px;}

.page-hard .main .midcolumn .canvabuttonline {display: flex; margin: 10px
auto 0; width: 400px;}

.page-hard .main .midcolumn .canvabuttonline a:first-child {margin-right:
auto;}

.page-hard .main .midcolumn .canvabuttonline a:not(:first-child) {margin-
left: 5px;}

.page-hard .main .rightcolumn {min-width: 350px; width: 30%; box-sizing:
border-box; flex: 0 0 30%;}

.page-hard .main .rightcolumn .container {background-color: #fff; margin: 0;
min-width: 350px; height: 490px;}

.page-hard .main .rightcolumn .container .code-reference {position:
relative;}

.page-hard .main .rightcolumn .container .code-reference .header {display:
flex; height:30px; align-items: center; flex-direction: row; width: 100%;
font-weight:bold; background-color: #f0f0f0; position: sticky; top: 0; z-
index: 1;}

.page-hard .main .rightcolumn .container .code-reference .scrollable
{overflow-y: scroll; max-height: 450px;}

```

```

.page-hard .main .rightcolumn .container .code-reference .row {display:
flex; flex-direction: row; width: 100%; margin-bottom: 10px; border-bottom:
1px solid #ccc;}

.page-hard .main .rightcolumn .container .code-reference .command {flex: 1;
font-weight: bold; min-width: 45%; margin-left: 5px;}

.page-hard .main .rightcolumn .container .code-reference .description {flex:
1; padding-left: 10px; min-width: 45%; margin-bottom: 10px;}

.page-hard .main .rightcolumn .container .code-reference .description
.example {margin-top: 10px;}

```

```

/* ---- Hard page Blocks ---- */

```

```

/* for Hard Page use

```

```

.page-medium .bottom .menu .block input {margin-left: 1em; width: 3em;
margin-bottom: 5px;}

```

```

.page-medium .main .script .block input {margin-left: 1em; width: 3em;
margin-bottom: 5px;}

```

```

.block.Back_to_center { background-image: url(../icons/Back_to_center.png);
background-size: 80px 80px; width: 80px; height: 80px;}
/* Only for the Back_to_center Block */

```

```

/* ----- Animations -----*/

```

```

@keyframes scale-up-center {

```

```

    0%{ -webkit-transform: scale(1); transform: scale(1); }

```

```

    50%{ -webkit-transform: scale(1.2); transform: scale(1.2); }

```

```

    100%{ -webkit-transform: scale(1); transform: scale(1); }

```

```

}

```

```

/* ----- Tooltips container ----- */

```

```

.tooltip { position: relative; display: inline-block;}

.tooltip .tooltiptext { visibility: hidden; width: 120px; background-color:
black; color: #fff; text-align: center; padding: 8px 12px; border-radius:
6px; font-size: 15px; position: absolute; z-index: 1; }

.tooltiptext.bottom_left {width: 120px; top: 30px; right: 30px; }
.tooltiptext.bottom_right {width: 180px; top: 30px; left: 30px; }
.tooltiptext.forbutton {width: 180px; bottom: 100px; left: -45px; }
/* Show the tooltip text when you mouse over the tooltip container */
.tooltip:hover .tooltiptext { visibility: visible;}

/* ----- Help Modal ----- */
.modal {
    display: none; position: fixed; /* Stay in place */
    z-index: 1; padding-top: 60px; left: 0; top: 0; width: 100%; height:
100%;
    overflow: auto; /* Enable scroll if needed */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */

    .modal-content {
        position: relative; background-color: #fefefe; margin: auto; padding:
0;
        border: 1px solid #888; width: 80%; box-shadow: 0 4px 8px 0
rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19);
        -webkit-animation-name: animatetop; -webkit-animation-duration: 0.4s;
animation-name: animatetop; animation-duration: 0.4s}

    @-webkit-keyframes animatetop { from {top:-300px; opacity:0} to {top:0;
opacity:1} }
    @keyframes animatetop { from {top:-300px; opacity:0} to {top:0; opacity:1} }

    .close { color: white; float: right; font-size: 28px; font-weight: bold;
}
    .close:hover, .close:focus { color: #000; text-decoration: none; cursor:
pointer;}

    .modal-header { padding: 16px 26px; background-color: #ff5301; color:
white; }
    .modal-body {padding: 16px 26px;}

```



```

.modal-footer { padding: 16px 26px; background-color: #ff5301; color: white;
}

.modal-body p {color:#3e3e3e; line-height: 1.4em; font-size: 16px;}

/* ----- Media Queries ----- */

@media only screen and (max-width:400px) {
  /* For mobile phones: */
  .left, .center, .right{ width:100%; }
}

/* ----- Arxeio ----- */

/*
div.top{width:100%;}
div.control-area {width:39%; border:1px solid #dcdcdc; background-
color:#dcdcdc; margin: 0px 0; padding: 5px; float: left;}
div.control-robot {padding:0; margin:0; float:left; }
div.script-column{ float: left; }
div.script-commands, div.script-examples{margin: 0 10px 10px 10px; padding:
5px; border: 3px #A8A8A8 solid; border-radius: 10px; -webkit-border-radius:
10px; -moz-border-radius: 10px;}
div.canvas-column{ width: 59%; height: 46%; float: right; margin: 0;
padding: 0; }
div.canvas-placeholder{ width: 100%; height: 100%; }
.canvas{ background-color: white; position: absolute; }

div.script-area{ background-color: #a8a8a8; padding: 15px 5px;}
div.script { border-radius: 5px; overflow-y: auto; padding:10px 6px;
background-color: #fff; }

div.menu-column{ padding: 10px 5px 20px 5px; background-color: #a8a8a8; }
p.menu-msg{text-align: center; padding: 5px; }
div.menu { border: 3px #c3c3c3 solid; border-radius: 10px; background-
color: PeachPuff; }

```

```

div.block { border: 1px inset #021429; border-radius: 7px; }
div.clear{width:100%; height:0px; float:none; clear:both;}
div.script div.block {float:left;}

div.menu div.block {display: inline-block; margin: 10px 5px;}

.script.over{ background-color: CadetBlue; }

.menu.over{ background-color: BurllyWood; }

.block{ padding: 5px; margin: 10px; cursor: move; }
.block.running{ outline: 2px solid red; }
.block.next{ outline: 2px solid green; }
.block.dragging{ opacity: 0.4; }
.menu .block{ background-color: Coral; }
.script .block{ margin: 0px; background-color: DodgerBlue; }
.block input{ margin-left: 1em; width: 3em; margin-bottom: 5px; }
.container{
    margin: 0 -6px 0 20px;
    padding-bottom: 10px;
    border: 1px inset #021429; border-radius: 7px;
    border-right: 0;
    background-color: LightSteelBlue;
    border-radius: 7px 0 0 7px;
}
.menu .container{ background-color: PeachPuff; }
.over .container{ background-color: CadetBlue; }

p.btn.btn-green {padding: 5px 0px; margin:0; width:100%;}
p.btn.rcv { border-color: #337ab7; padding: 5px 0px; margin:0; width:100%;}
input.btn.rcv {border-color: #337ab7; padding: 5px 0px; margin:0;
width:100%;}

div.script-column .btn{display:block; width: 90%; margin: 5px auto;
padding: 5px;}

```

```

.block.Repeat { background-image: url(/icons/Repeat.svg);    background-
repeat: no-repeat;}

.block.Speed { background-image: url(/icons/Speed.svg);    background-repeat:
no-repeat;}

.block.Stop { background-image: url(/icons/Stop.svg);    background-repeat:
no-repeat;}

.block.Forward {background-image: url(/icons/Forward.svg);    background-
repeat: no-repeat;}

.block.Backward{background-image: url(/icons/Backward.svg);    background-
repeat: no-repeat;}

.block.Left{background-image: url(/icons/Left.svg);    background-repeat: no-
repeat;}

.block.Right{background-image: url(/icons/Right.svg);    background-repeat:
no-repeat;}

.block.RotateCW{background-image: url(/icons/RotateCW.svg);    background-
repeat: no-repeat;}

.block.RotateCCW{background-image: url(/icons/RotateCCW.svg);    background-
repeat: no-repeat;}

.block.FwdRight{background-image: url(/icons/FwdRight.svg);    background-
repeat: no-repeat;}

.block.FwdLeft{background-image: url(/icons/FwdLeft.svg);    background-
repeat: no-repeat;}

.block.BwdRight{background-image: url(/icons/BwdRight.svg);    background-
repeat: no-repeat;}

.block.BwdLeft{background-image: url(/icons/BwdLeft.svg);    background-
repeat: no-repeat;}

.block.Speed{background-image: url(/icons/Speed.svg);    background-repeat:
no-repeat;}

.block.Delay{background-image: url(/icons/Delay.svg);    background-repeat:
no-repeat;}

.block.Pen_up{background-image: url(/icons/Pen_up.svg);    background-repeat:
no-repeat;}

.block.Pen_down{background-image: url(/icons/Pen_down.svg);    background-
repeat: no-repeat;}

.block.Home{background-image: url(/icons/Home.svg);    background-repeat: no-
repeat;}

.block.Hide_turtle{background-image:    url(/icons/Hide_turtle.svg);
background-repeat: no-repeat;}

.block.Show_turtle{background-image:    url(/icons/Show_turtle.svg);
background-repeat: no-repeat;}

```

## Project/data/js/custom-cmd.js

```
var codeVerifier = true;

// Define the outf function for the output of the code
function outf(text) {
    var mypre = document.getElementById("output");
    mypre.innerHTML = mypre.innerHTML + text;
}

// Define builtinRead function - library throws error for some reason
function builtinRead(x) {
    if (Sk.builtinFiles === undefined || Sk.builtinFiles["files"][x] ===
        undefined) throw "File not found: '" + x + "'";
    return Sk.builtinFiles["files"][x];
}

// Function to match the command pattern from codemirror and translate to
// turtle commands
function interpretCommand(command) {
    var match = command.match(/(\w+)\s*(([^\s]+)\s)/);
    if (match) {
        var commandType = match[1];
        var parameter = parseFloat(match[2]);
        switch (commandType) {
            case 'Forward':
                return 't.forward(' + parameter + ')';
            case 'ForwardLeft':
                return 't.left(45)\nt.forward(' + parameter + ')';
            case 'ForwardRight':
                return 't.right(45)\nt.forward(' + parameter + ')';
            case 'RotateCCW':
                return 't.left(' + parameter + ')';
            case 'Left90':
                return 't.left(90)';
            case 'RotateCW':
                return 't.right(' + parameter + ')';
            case 'Right90':
                return 't.right(90)';
            case 'Backward':
```

```

        return 't.backward(' + parameter + ')';
    case 'BackwardLeft':
        return 't.right(45)\nt.backward(' + parameter + ')';
    case 'BackwardRight':
        return 't.left(45)\nt.backward(' + parameter + ')';
    case 'Speed':
        if (parameter < 0) {
            return 't.speed(1)';
        } else if (parameter <= 1023) {
            var mappedSpeed = Math.ceil(parameter / 102.3);
            if (mappedSpeed < 1) {
                return 't.speed(1)';
            } else if (mappedSpeed > 10) {
                return 't.speed(10)';
            } else {
                return 't.speed(' + mappedSpeed + ')';
            }
        } else {
            return 't.speed(0)';
        }
    default:
        return command;
    }
} else {
    return console.log("Invalid command format:" + command);
}
}

// Function to handle the repetition logic
function repeatLogic(commandLines, startIndex) {
    var repeatLine = commandLines[startIndex];
    var repeatMatch = repeatLine.match(/Repeat\((\d+)\)/);
    if (!repeatMatch) {
        console.log("Invalid Repeat command format:", repeatLine);
        return;
    }
    var repeatCount = parseInt(repeatMatch[1]);
    var interpretedCode = '';

```

```

    for (var i = startIndex + 1; i < commandLines.length; i++) {
        var line = commandLines[i];
        if (line.trim() === '{}') {
            break; // End of Repeat block
        }
        interpretedCode += interpretCommand(line) + '\n';
    }
    // Construct repeated commands
    var repeatedCommands = '';
    for (var j = 0; j < repeatCount; j++) {
        repeatedCommands += interpretedCode;
    }
    return repeatedCommands;
}
// Function to run code on Skulpt turtle mode
function runCode() {
    var code = editor.getValue();
    var animationStatus = $('animate-action').data('status');
    var tracerValue = (animationStatus === 'on') ? 1 : 0;
    var commandLines = code.split('\n');
    var interpretedCode = '';
    for (var i = 0; i < commandLines.length; i++) {
        var line = commandLines[i];
        if (line.trim().startsWith('Repeat')) {
            var repeatLogicCode = repeatLogic(commandLines, i);
            interpretedCode += repeatLogicCode;
            // Skip lines inside Repeat block
            while (i < commandLines.length &&
!commandLines[i].includes('{}')) {
                i++;
            }
        } else {
            interpretedCode += interpretCommand(line) + '\n';
        }
    }
    // Construct modifyCode
    var modifyCode = "import turtle\n" + "turtle.tracer(" + tracerValue +
")\n" + "t = turtle.Turtle()\n" + interpretedCode;

```

```

    if (animationStatus !== 'on') {
        modifyCode += "\nturtle.update()";
    }

    console.log("Code to be executed:", modifyCode);
    // Execute code using Skulpt
    var mypre = document.getElementById("output");
    mypre.innerHTML = '';
    Sk.pre = "output";
    Sk.configure({output:outf, read:builtinRead});
    (Sk.TurtleGraphics || (Sk.TurtleGraphics = {})).target = 'canvas';
    Sk.misceval.asyncToPromise(function() {
        return Sk.importMainWithBody("<stdin>", false, modifyCode, true);
    }).then(function(mod) {
        console.log('Code executed successfully');
    }, function(err) {
        console.log('Error executing code:', err.toString());
    });
}

// Function to match the command pattern from codemirror and translate to
// turtle commands
function interpretEspCommand(command, speed) {
    var match = command.match(/(\w+)\(((^)+)\)/);
    if (match) {
        var commandType = match[1];
        var parameter = parseFloat(match[2]); //The distance is parsed as
        parameter but is not implemented. I need to figure out how it will relate to
        actual moves.

        switch (commandType) {
            case 'Forward':
            case 'ForwardLeft':
            case 'ForwardRight':
            case 'RotateCCW':
            case 'Left90':
            case 'RotateCW':
            case 'Right90':
            case 'Backward':
            case 'BackwardLeft':
            case 'BackwardRight':

```

```

        case 'Speed':
            return [commandType, speed];
        default:
            console.log("Invalid command format:" + command);
            codeVerifier = false;
            return [];
    }
} else {
    console.log("Invalid command format:" + command);
    codeVerifier = false;
    return [];
}
}

// Function to handle the repetition logic
function repeatEspLogic(commandLines, startIndex, speed) {
    var repeatLine = commandLines[startIndex];
    var repeatMatch = repeatLine.match(/Repeat\((\d+)\)/);
    if (!repeatMatch) {
        console.log("Invalid Repeat command format:", repeatLine);
        return [];
    }
    var repeatCount = parseInt(repeatMatch[1]);
    var interpretedCode = [];
    for (var i = startIndex + 1; i < commandLines.length; i++) {
        var line = commandLines[i];
        if (line.trim() === '') {
            break; // End of Repeat block
        }
        var interpretedCommand = interpretEspCommand(line, speed);
        if (interpretedCommand !== null) {
            interpretedCode.push(interpretedCommand);
        }
    }
    // Construct repeated commands
    var repeatedCommands = [];
    for (var j = 0; j < repeatCount; j++) {
        repeatedCommands.push.apply(repeatedCommands, interpretedCode);
    }
}

```



```

    }
    return repeatedCommands;
}

function runESP() {
    var code = editor.getValue();
    var commandLines = code.split('\n');
    var espCommands = [];
    var speed = 512;
    var codeVerifier = true;
    // check for speed value
    commandLines.forEach(function(line) {
        if (line.trim().startsWith('Speed')) {
            var match = line.match(/Speed\(((^)+)\)/);
            if (match) {
                var userSpeed = parseFloat(match[1]);
                speed = Math.max(1, Math.min(1023, userSpeed));
                return;
            }
        }
    });
    for (var i = 0; i < commandLines.length; i++) {
        var line = commandLines[i];
        if (!codeVerifier) {
            break;
        }
        if (line.trim().startsWith('Repeat')) {
            var repeatLogicCode = repeatEspLogic(commandLines, i, speed);
            espCommands.push.apply(espCommands, repeatLogicCode);
            // Skip lines inside Repeat block
            while (i < commandLines.length &&
!commandLines[i].includes(' ')) {
                i++;
            }
        } else {
            var interpretedCommand = interpretEspCommand(line, speed);
            if (interpretedCommand !== null &&
interpretedCommand.length > 0) {
                espCommands.push(interpretedCommand);
            }
        }
    }
}

```

```

        } else {
            codeVerifier = false;
            break;
        }
    }
}
if (codeVerifier && espCommands.length > 0) {
    var espData = JSON.stringify(espCommands);
    console.log("**** espData = " + espData);
    connection.send(espData);
} else {
    console.log("Invalid command on Editor Detected. Voiding
attempt to reach receiver.");
}
}
function cmdRun() {
    var command = $('#commandInput').val().trim();
    var codeVerifier = true;
    if (!command) {
        return;
    } else if (command === 'PathPlan') {
        $('#commandInput').val('');
        runESP();
    } else {
        var interpretedCommand = interpretEspCommand(command, speed);
        if (codeVerifier && interpretedCommand.length > 0) {
            var espData = JSON.stringify([interpretedCommand]);
            $('#commandInput').val('');
            console.log("**** espData = " + espData);
            connection.send(espData);
        } else {
            console.log("Command on terminal not recognized:",
command);
        }
    }
}
}

```