



ΠΟΛΥΤΕΧΝΕΙΟ  
ΚΡΗΤΗΣ /  
**TECHNICAL  
UNIVERSITY  
OF CRETE**

ΣΧΟΛΗ  
**ΗΛΕΚΤΡΟΛΟΓΩΝ  
ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ**

Διπλωματική Εργασία

# **Ανάπτυξη Ηλεκτρονικού Συστήματος για Εκπαίδευση STEM**

**Χρυσή Γιαννουλάκη**

ΑΜ: 2015030202

**Εξεταστική επιτροπή:**

**Καθηγητής Ευτύχιος Κουτρούλης (Επιβλέπων)**

**Καθηγητής Μιχαήλ Λαγουδάκης**

**Αναπλ. Καθηγητής Ελευθέριος Δοϊτσίδης (Σχολή ΜΠΔ)**

Χανιά 2024

## ΠΕΡΙΛΗΨΗ

Το τελευταίο χρονικό διάστημα παρατηρείται αύξηση ενδιαφέροντος σχετικά με την εκπαίδευση STEM (Science, Technology, Engineering and Mathematics). Βαθμιαία, μαζί με την εκπαιδευτική ρομποτική, ενσωματώνεται τόσο στα προγράμματα σπουδών των σχολείων της Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης, όσο και σε εκπαιδευτικές εξωσχολικές δραστηριότητες. Αυτή η εκπαιδευτική προσέγγιση είναι πολύ σημαντική και παρόλο που έχει ευρέως υιοθετηθεί τις τελευταίες δεκαετίες εξακολουθεί να χρειάζεται προσιτά εργαλεία με εκτεταμένες δυνατότητες. Ωστόσο, τα υπάρχοντα σχετικά συστήματα αντιμετωπίζουν αρκετές προκλήσεις, συμπεριλαμβανομένων του συνολικού κόστους και της πολυπλοκότητας του συνολικού συστήματος, καθιστώντας επιτακτική την ανάγκη εξεύρεσης άλλων συστημάτων εκμάθησης, χωρίς τα μειονεκτήματα αυτά. Στην παρούσα διπλωματική εργασία, αναπτύχθηκε το σύστημα ColourBot, το οποίο αποτελεί μια λύση χαμηλού κόστους για εκπαίδευση STEM. Περιλαμβάνει προσαρμοσμένο λογισμικό (software) και υλικό (hardware) που περικλείεται από λειτουργικά κελύφη. Τα κελύφη αυτά συνδέονται χρησιμοποιώντας έναν προσαρμοσμένο μηχανισμό σύνδεσης με ενισχυμένες δυνατότητες, ο οποίος επιτρέπει τον σχεδιασμό και την ανάπτυξη διάφορων λειτουργικών πρωτοτύπων. Το σύστημα παρέχει προσαρμοσμένο λογισμικό ανάλογα με την ηλικία, το εκπαιδευτικό επίπεδο και τις δεξιότητες των χρηστών, οι οποίοι στην περίπτωση μας είναι μαθητές της Πρωτοβάθμιας και Δευτεροβάθμιας εκπαίδευσης. Ο προγραμματισμός του υλικού διαφοροποιείται από τη συνηθισμένη ανάπτυξη συμβατικού κώδικα. Βασίζεται σε περιβάλλον οπτικού προγραμματισμού (visual programming), δίνοντας όμως την δυνατότητα στον χρήστη να βλέπει και τον πραγματικό κώδικα. Τα πλεονεκτήματα του είναι ότι αποτελείται από ηλεκτρονικά χαμηλού κόστους και μικρού μεγέθους, μια συνδεσμολογία που ελαχιστοποιεί την πολυπλοκότητα και την αστοχία σύνδεσης, περίβλημα αρκετά στιβαρό το οποίο είναι παράλληλα εύκολο στη χρήση και μια φιλική πλατφόρμα λογισμικού κατάλληλα σχεδιασμένη. Κατά τη συγκεκριμένη προσέγγιση αναπτύχθηκαν νέα αρθρώματα και αναπτύχθηκαν συγκεκριμένα μαθησιακά σενάρια. Αυτός ο σχεδιασμός δίνει τη δυνατότητα για την κατασκευή πρωτότυπων ρομποτικών συσκευών και συστημάτων αυτοματισμού, ενισχύοντας τη δημιουργική σκέψη των μαθητών.

## ABSTRACT

Recently, there has been an increasing interest regarding STEM education (Science, Technology, Engineering, and Mathematics). Along with educational robotics, STEM is gradually being integrated into the curricula of Primary and Secondary Education schools, as well as in extracurricular educational activities. This educational approach is very important, and although it has been widely adopted in the past decades, it still requires affordable tools with extensive capabilities. However, existing related systems face several challenges, including overall cost and system complexity, making it crucial to find alternative learning systems without these drawbacks. In this diploma, the ColourBot system was developed, offering a low-cost solution for STEM education. It includes custom software and hardware, enclosed in functional shells. These shells are connected using a custom connection mechanism with enhanced capabilities, allowing for the design and development of various functional prototypes. The system provides customized software tailored to the age, educational level, and skills of the users, who, in this case, are students in Primary and Secondary Education. The programming of the hardware differs from the usual development of conventional code. It is based on a visual programming environment, but it also allows the user to view the actual code. Its advantages include electronics of low cost and small size, a connection system that minimizes complexity and connection failures, a robust casing that is also easy to use, and a user-friendly software platform that is appropriately designed. Through this approach, new modules were developed, and specific learning scenarios were created. This particular design allows for the creation of prototype robotic devices and automation systems, encouraging students' creative thinking.

## Πίνακας περιεχομένων

---

<b>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ .....</b>	<b>7</b>
1.1 Διάρθρωση διπλωματικής εργασίας .....	8
1.2 Στόχος της παρούσας εργασίας .....	8
<b>ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ STEM ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....</b>	<b>12</b>
2.1 Εκπαίδευση STEM .....	12
2.2 Εκπαιδευτική ρομποτική .....	13
2.3 Εκπαιδευτικές πλατφόρμες .....	16
2.4 Υφιστάμενα λογισμικά εκπαιδευτικής ρομποτικής .....	19
2.4.1 Scratch.....	19
2.4.2 Snap! .....	21
2.4.3 MakeCode .....	22
2.4.4 Blockly .....	23
2.5 Προκλήσεις .....	24
<b>ΚΕΦΑΛΑΙΟ 3: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ HARDWARE ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>26</b>
3.1 Σχεδιαστική φιλοσοφία.....	26
3.2 Σύνδεση δομικών στοιχείων .....	29
3.2.1 Γενικές πληροφορίες των USB Type-C .....	29
3.2.2 Εφαρμογή της Type-C συνδεσμολογίας .....	31
3.3 Κεντρική Μονάδα Συστήματος .....	32
3.4 Αισθητήρες και Επενεργητές .....	35
3.4.1 Άρθρωμα LED.....	36
3.4.2 Άρθρωμα DC Motor .....	37
3.4.3 Άρθρωμα LCD.....	40
3.4.4 Άρθρωμα Buttons .....	42
3.4.5 Άρθρωμα Sonar Sensor.....	43
3.4.6 Άρθρωμα Infrared Sensor .....	45
3.4.7 Άρθρωμα Temperature & Humidity Sensor.....	47

3.4.8 Άρθρωμα Line Tracking .....	48
3.4.9 Άρθρωμα Joystick .....	50
<b>3.5 Μονάδα Ισχύος .....</b>	<b>51</b>
<b>3.6 Πρωτότυπες πλακέτες .....</b>	<b>54</b>
<b>ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ArduBlockly.....</b>	<b>55</b>
4.1 Google Blockly.....	55
4.2 Ardublockly .....	56
4.3 Εργαλείο Block Factory .....	57
4.4 Εργαλείο Block Exporter .....	59
4.5 Εργαλείο Workspace Factory .....	60
4.6 Ορισμός μπλοκ.....	61
4.6.1 Συνδεσιμότητα .....	61
4.6.2 Τύποι μπλοκ .....	62
4.7 Διαδικασία αρχικοποίησης για την ανάπτυξη του Ardublockly.....	63
4.8 Διαδικασία ανάπτυξης του Ardublockly .....	65
<b>ΚΕΦΑΛΑΙΟ 5: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>71</b>
5.1 Προσαρμογή του λογισμικού Ardublockly στις ανάγκες της εκπαιδευτικής πλατφόρμας .....	71
5.2 Μπλοκ της κατηγορίας ColourBot .....	74
5.2.1 Μπλοκ Sonar Sensor .....	74
5.2.2 Μπλοκ Temperature & Humidity.....	76
5.2.3 Μπλοκ Infrared Sensor .....	78
5.2.4 Μπλοκ Line Tracking .....	79
5.2.5 Μπλοκ LED .....	81
5.2.6 Μπλοκ Buttons.....	83
5.2.7 Μπλοκ LCD .....	86
5.2.8 Μπλοκ Joystick.....	88
5.2.9 Μπλοκ DC Motor .....	89
<b>ΚΕΦΑΛΑΙΟ 6: ΤΡΙΣΔΙΑΣΤΑΤΑ ΚΕΛΥΦΗ.....</b>	<b>92</b>
6.1 Τρισδιάστατα κελύφη συστήματος .....	92
6.2 Μηχανισμός σύνδεσης .....	95

<b>ΚΕΦΑΛΑΙΟ 7: ΕΦΑΡΜΟΓΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ .....</b>	<b>97</b>
<b>7.1 Ενδεικτικές εφαρμογές.....</b>	<b>97</b>
7.1.1 Ρυθμιζόμενα LED από κουμπιά .....	97
7.1.2 Εμφάνιση δεδομένων στην οθόνη.....	99
7.1.3 Ανίχνευση εμποδίου .....	101
7.1.4 Ρομποτικό όχημα .....	103
<b>7.2 Μαθησιακά σενάρια .....</b>	<b>106</b>
<b>ΚΕΦΑΛΑΙΟ 8: ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>107</b>
8.1 Συμπεράσματα.....	107
8.2 Μελλοντικές Επεκτάσεις.....	108
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>109</b>
<b>Παράρτημα 1: Φύλλα Εργασίας και Ερωτηματολόγια .....</b>	<b>111</b>

## ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

---

Η εκπαίδευση με τη προσέγγιση STEM (Science, Technology, Engineering and Mathematics) αντιπροσωπεύει ένα σύγχρονο παράδειγμα εκπαίδευσης που βασίζεται σε προηγμένες θεωρίες και προσεγγίσεις, επιδιώκοντας την ολοκληρωμένη ανάπτυξη μαθητών στα πεδία της Επιστήμης, της Τεχνολογίας, της Μηχανικής και των Μαθηματικών. Διαμορφώνει μαθητές έτοιμους να προσαρμοστούν στις προκλήσεις του σύγχρονου κόσμου με επιστημονική, τεχνολογική και δημιουργική κατάρτιση. Οι μαθητές έχουν την δυνατότητα να βελτιώσουν την κριτική τους σκέψη και την ικανότητα τους να επιλύουν προβλήματα. Παράλληλα, κάποιοι έχουν αναφέρει θετική επίδραση στην αυτοβελτίωση, συμπεριλαμβανομένων της συνεργασίας, των κοινωνικών και επικοινωνιακών δεξιοτήτων, των γνωστικών και των κοινωνικών ευθυνών [1].

Η έννοια του όρου STEM Education αφορά τη διδασκαλία και την εκμάθηση σε τέσσερα βασικά πεδία και είναι προσανατολισμένη προς κάθε ηλικιακή κατηγορία, από την προσχολική ηλικία μέχρι τη μεταδιδακτορική βαθμίδα. Η προσέγγιση STEM βασίζεται στη θεωρία του εποικοδομητισμού (constructionism learning theory), που αντλεί έμπνευση από τη θεωρία του κονστρουκτιβισμού (constructivism theory). Η συγκεκριμένη προσέγγιση, προάγει ότι η μάθηση επιτυγχάνεται καλύτερα μέσα από τον σχεδιασμό και την υλοποίηση έργων από τους μαθητές [2]. Συνολικά, οι μαθητές συμμετέχουν σε εργασίες ή έργα, όπου προωθείται η ομαδική συνεργασία, και ο εκπαιδευτικός υιοθετεί έναν ρόλο καθοδηγητή, κρίνοντας πότε είναι αναγκαία η παρέμβασή του. Αυτή η πρακτική έρχεται σε πλήρη αντίθεση με τις παραδοσιακές μεθόδους διδασκαλίας, οι οποίες εστιάζουν κυρίως σε διαλέξεις με συνεχή καθοδήγηση και παρέμβαση από τον διδάσκοντα. Αυτές οι προσεγγίσεις κερδίζουν όλο και περισσότερη αποδοχή, ενσωματώνονται στα σύγχρονα σχολικά προγράμματα σε όλα τα επίπεδα εκπαίδευσης, και επεκτείνονται σε εξωσχολικές εκπαιδευτικές δραστηριότητες.

Η παρούσα εργασία συμμορφώνεται με τις απαιτήσεις των σύγχρονων προγραμμάτων σπουδών, παρουσιάζει ένα νέο εκπαιδευτικό εργαλείο προσανατολισμένο στις ανάγκες του STEM στην Πρωτοβάθμια και Δευτεροβάθμια εκπαίδευση, χρησιμοποιώντας ανοικτά και χαμηλού κόστους εργαλεία. Το σύστημα ColourBot, το οποίο αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας, είναι μία ολοκληρωμένη εκπαιδευτική πλατφόρμα η οποία

αποτελείται από προσαρμοσμένο υλικό και λογισμικό και μπορεί να χρησιμοποιηθεί στη διδασκαλία εννοιών της πληροφορικής και βασικών αυτοματισμών.

## ***1.1 Διάρθρωση διπλωματικής εργασίας***

Η παρούσα διπλωματική εργασία δομείται ως εξής:

Στο Κεφάλαιο 2, γίνεται μια σύντομη παρουσίαση της μεθοδολογίας STEM και παρουσιάζονται οι υπάρχουσες διαθέσιμες λύσεις στον τομέα.

Στο Κεφάλαιο 3 παρουσιάζεται λεπτομερώς η διαδικασία σχεδιασμού των ηλεκτρονικών τμημάτων της πλατφόρμας, περιλαμβάνοντας τα διαγράμματα και τα σχέδια των τυπωμένων κυκλωμάτων. Περιγράφεται αναλυτικά η σχεδιαστική προσέγγιση που ακολουθήθηκε από την αρχική φάση μέχρι την ολοκλήρωση του τελικού πρωτοτύπου.

Στα Κεφάλαια 4 και 5 αναλύονται διεξοδικά οι δυνατότητες του εργαλείου Ardublockly, που αποτέλεσε το βασικό εργαλείο για την δημιουργία των δομικών στοιχείων του λογισμικού, καθώς και οι αλλαγές και προσθήκες που πραγματοποιήθηκαν στην αρχική έκδοση του ώστε να είναι δυνατή η χρήση των πρωτότυπων συσκευών μέσω αυτού.

Το Κεφάλαιο 6 περιλαμβάνει τα εξωτερικά κελύφη για τα αρθρώματα της πλατφόρμας, τα τρισδιάστατα μοντέλα καθώς και φωτογραφίες.

Στο Κεφάλαιο 7 παρουσιάζονται μαθησιακά σενάρια ώστε να γίνει πλήρως η πρώτη πιλοτική χρήση της πλατφόρμας από μαθητές, καθώς και κάποια ενδεικτικά παραδείγματα χρήσης.

Στο τελευταίο κεφάλαιο της εργασίας (Κεφάλαιο 8) γίνεται μια συνοπτική ανασκόπηση όλων όσα έχουν αναφερθεί προηγουμένως, καθώς και τα συμπεράσματα που προκύπτουν. Επιπλέον, παρατίθενται κάποιες σκέψεις και προτάσεις για μελλοντικές επεκτάσεις και σχεδιαστικές αλλαγές της πλατφόρμας.

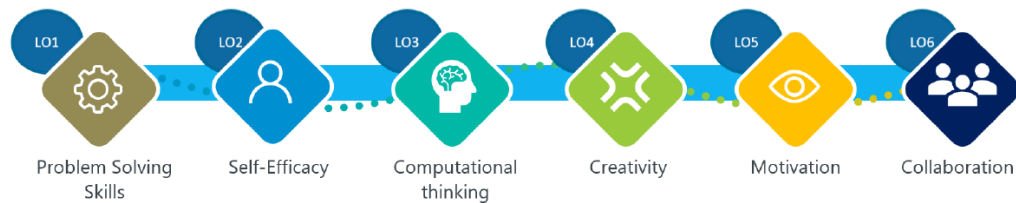
Τέλος στο Παράρτημα 1 παρατίθενται τα φύλλα έργου που χρησιμοποιήθηκαν στην επίδειξη του συστήματος σε μαθητές.

## ***1.2 Στόχος της παρούσας εργασίας***

Στην παρούσα εργασία αναπτύσσεται ένα ηλεκτρονικό σύστημα για εκπαίδευση STEM που ενσωματώνει πολλά και διαφορετικά λειτουργικά χαρακτηριστικά, αλλά παράλληλα διατηρεί χαμηλό κόστος που αποτελεί και συγκριτικό πλεονέκτημα, σε σχέση με τις λύσεις που υπάρχουν στη βιβλιογραφία μέχρι τώρα. Έξι μαθησιακά αποτελέσματα έχουν εντοπιστεί στη



βιβλιογραφία ως θεμελιώδεις πυλώνες για τον σχεδιασμό και τη δημιουργία αποτελεσματικών εκπαιδευτικών δραστηριοτήτων σχετικών με το STEM, όπως έχουν αναγνωριστεί και αναλυθεί με λεπτομέρεια στην εργασία της Enripidou *et al.* [3] και παρουσιάζονται στην Εικόνα 1.2.



Εικόνα 1-2. Αναμενόμενα Μαθησιακά Αποτελέσματα [3].

Το προτεινόμενο σύστημα βασίζεται στην προσπάθεια επίτευξης των παραπάνω μαθησιακών αποτελεσμάτων και συγκεκριμένα:

**Ικανότητα επίλυσης προβλημάτων (Problem-solving skills):** Αναφέρεται στις δραστηριότητες που επιτρέπουν στους μαθητές να αναπτύξουν δεξιότητες στη μηχανική, στην διαχείριση καταστάσεων, στον εντοπισμό των αιτιών και στην αναζήτηση λύσεων.

**Αυτοαποτελεσματικότητα (Self-efficacy):** Η αυτοαποτελεσματικότητα συνδέεται με την πίστη του ατόμου στις δικές του ικανότητες να επιτύχει σε συγκεκριμένες καταστάσεις ή καθήκοντα. Δεν αντικατοπτρίζει αναγκαστικά τις πραγματικές του δυνατότητες αλλά επηρεάζει σημαντικά την προσπάθεια του απέναντι σε νέες προκλήσεις. Οι δραστηριότητες θα πρέπει να σχεδιάζονται με στόχο την ενίσχυση αυτής της πίστης στους μαθητές

**Υπολογιστική σκέψη (Computational thinking):** Η υπολογιστική σκέψη αναφέρεται στον τρόπο με τον οποίο οι μαθητές επεξεργάζονται τα δεδομένα ενός προβλήματος, αναγνωρίζουν τις σημαντικές παραμέτρους του και οργανώνουν τα βήματα για την επίλυσή του. Η εκπαιδευτική ρομποτική συμβάλλει στην ανάπτυξη αυτής της σκέψης μέσω της κατάλληλης οργάνωσης των δραστηριοτήτων.

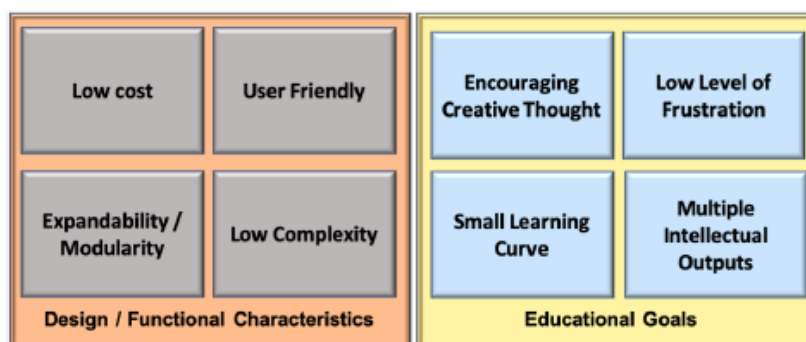
**Δημιουργικότητα (Creativity):** Οι δραστηριότητες που σχετίζονται με την εκπαιδευτική ρομποτική πρέπει να ενθαρρύνουν τη δημιουργικότητα των μαθητών, ώστε να σκέφτονται έξω από τα καθιερωμένα πλαίσια και να αναπτύσσουν νέες, πρωτότυπες λύσεις ή ιδέες χρησιμοποιώντας καινοτόμες προσεγγίσεις.

**Κίνητρα (Motivation):** Κάθε δραστηριότητα θα πρέπει να προσφέρει ισχυρά κίνητρα στους μαθητές, καθώς αποτελούν καθοριστικό παράγοντα για την επίτευξη των εκπαιδευτικών στόχων, επιβραβεύοντας τόσο το τελικό αποτέλεσμα όσο και τη διαδικασία που ακολουθείται για την επίτευξή του. Πρέπει να είναι ενδιαφέρουσες και να προσφέρουν προκλήσεις που είναι

εντός των δυνατοτήτων τους, ενώ ταυτόχρονα πρέπει να ενθαρρύνουν την αυτονομία τους και να ενισχύουν την αίσθηση της επίτευξης.

**Συνεργασία (Collaboration):** Η συνεργασία είναι απαραίτητη για την επιτυχία στην εκπαιδευτική ρομποτική, καθώς οι μαθητές συχνά καλούνται να εργαστούν σε ομάδες. Η ανάπτυξη πολύπλοκων λειτουργικών συσκευών ενθαρρύνει, λόγω της φύσης των δραστηριοτήτων της εκπαιδευτικής ρομποτικής, την ομαδικότητα και τη συνεργασία.

Το σύστημα που αναπτύχθηκε ακολουθεί την λογική ανάπτυξης που παρουσιάστηκε στις εργασίες [4, 5] υιοθετώντας μια ολιστική προσέγγιση για την επίτευξη των στόχων, αλλά διαφοροποιείται σημαντικά. Γίνεται προσπάθεια να αντιμετωπιστούν συγκεκριμένα σχεδιαστικά και λειτουργικά χαρακτηριστικά, ενώ ταυτόχρονα επιδιώκει να καλύψει όσον το δυνατόν περισσότερους εκπαιδευτικούς στόχους. Όσον αφορά τα σχεδιαστικά και λειτουργικά χαρακτηριστικά, η πλατφόρμα ιδανικά πρέπει να έχει χαμηλό κόστος, να είναι φιλική προς τον χρήστη, να έχει χαμηλή πολυπλοκότητα, καθώς και να είναι αρθρωτή και επεκτάσιμη. Όσον αφορά τους εκπαιδευτικούς στόχους, πρέπει η πλατφόρμα να ενισχύει τη δημιουργική σκέψη των μαθητών και να έχει μια μικρή καμπύλη μάθησης που να διατηρεί χαμηλά τα επίπεδα απογοήτευσης για τα παιδιά. Όλα αυτά πρέπει να συνδυαστούν με τρόπο ώστε η πλατφόρμα να τους επιτρέπει να επιτύχουν τα βασικά μαθησιακά αποτελέσματα όπως αυτά έχουν περιγραφεί. Η ανάπτυξη έχει πραγματοποιηθεί με τη χρήση ανοικτών εργαλείων στοχεύοντας στη μείωση του κόστους αλλά παράλληλα και στην ενίσχυση της συλλογικότητας μέσω της κοινότητας που υποστηρίζει τα συγκεκριμένα εργαλεία.



Εικόνα 1-3. Σχεδιαστική Φιλοσοφία [5].

Μέρος της παρούσας εργασίας έχει παρουσιαστεί στην ακόλουθη εργασία:

**C. Giannoulaki, A. Sioutis, G. Kypraios, G. Tsinarakis and L. Doitsidis, "ColourBot: A New Modular Tool for STEM Education," 2024 32<sup>nd</sup> Mediterranean Conference on Control and**

*Automation (MED)*, Chania - Crete, Greece, 2024, pp. 400-405, doi:  
10.1109/MED61351.2024.10566228.

## ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ STEM ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

---

### *2.1 Εκπαίδευση STEM*

Η εκπαίδευση STEM αποτελεί ένα σημαντικό μέρος της σύγχρονης εκπαιδευτικής προσέγγισης. Από την απλή κατασκευή μοντέλων έως την ανάπτυξη προηγμένων ρομποτικών συστημάτων, η εκπαίδευση STEM συμβάλλει στην κατανόηση της φύσης και της λειτουργίας του κόσμου γύρω μας. Ενισχύει την ενασχόληση με τον προγραμματισμό, τους αυτοματισμούς και τη ρομποτική, μεταξύ άλλων. Αναμφίβολα, ο καθοριστικός ρόλος του STEM έχει σημαντική βαρύτητα, επηρεάζοντας τις διδακτικές μεθόδους σε όλα τα επίπεδα εκπαίδευσης [6, 7]. Σύμφωνα με τα ευρήματα που παρουσιάζονται από τους συγγραφείς στο [8], τα χαρακτηριστικά της εκπαίδευσης με επίκεντρο το STEM περιλαμβάνουν μαθησιακές εμπειρίες που ενσωματώνουν πολλαπλά γνωστικά πεδία. Οι μαθητές έχουν την ευκαιρία να αναβαθμίσουν την κριτική τους σκέψη και τις δεξιότητές τους στην επίλυση προβλημάτων, και κάποιοι αναφέρουν θετική επίδραση στην προσωπική τους ανάπτυξη, όπως στη συνεργασία, στις κοινωνικές, επικοινωνιακές και μεταγνωστικές δεξιότητες, καθώς και στις γνωστικές, κοινωνικές τους ευθύνες [9]. Μέσω της ανάπτυξης αυτοπεποίθησης και αυτοκαθοδήγησης, οι μαθητές ενισχύουν τη δημιουργικότητα, την καινοτομία και την κινητοποίησή τους [6], [10]. Ο κύριος στόχος, ωστόσο, είναι οι νέοι να αποκτήσουν το απαραίτητο υπόβαθρο στις επιστήμες, την τεχνολογία, τη μηχανική και τα μαθηματικά, προκειμένου να είναι σε θέση να ανταπεξέλθουν στις επαγγελματικές προκλήσεις των μελλοντικών ετών.

Ο όρος πρωτοεμφανίστηκε ως «SMET» στις Ηνωμένες Πολιτείες κατά τη δεκαετία του 1990. Το National Science Foundation (NSF) της Αμερικής επιδίωξε να δώσει ιδιαίτερη έμφαση σε κάθε μία από τις τέσσερις πτυχές του, με δύο κύριους στόχους: (α) να ενισχύσει τις τεχνολογικές και μηχανικές εξελίξεις, προκειμένου η χώρα να παραμείνει ανταγωνιστική σε παγκόσμιο επίπεδο και (β) να εξασφαλίσει ότι κάθε μαθητής θα κατανοεί και θα γνωρίζει τις βασικές αρχές και μεθόδους των μαθημάτων SMET, καθώς και τη σύνδεση μεταξύ τους, ώστε το μελλοντικό ανθρώπινο δυναμικό να διαθέτει τα απαραίτητα εφόδια. Το 1998,

πραγματοποιήθηκε η πρώτη προσπάθεια για την προετοιμασία των κατάλληλων συνθηκών που θα εξασφαλίσουν την ανάπτυξη και τη διατήρηση των ικανοτήτων στους τομείς των επιστημών, των μαθηματικών, της μηχανικής και της τεχνολογίας. Έτσι, ο όρος SMET καθιερώθηκε, μέχρι που το 2001 αντικαταστάθηκε από το σημερινό STEM (Science, Technology, Engineering, Mathematics) από τη βιολόγο και διευθύντρια του NSF, Judith A. Ramaley.

Τα τελευταία χρόνια έχουν κάνει την εμφάνιση τους κάποιες παραλλαγές. Το STEAM (Science, Technology, Engineering, the Arts and Mathematics) υιοθετεί πλήρως την διδακτική μεθοδολογία του STEM και προσθέτει επιπλέον την συνιστώσα της Τέχνης. Η βασική διαφορά μεταξύ του STEM και του STEAM είναι ότι το STEM εστιάζει σε έννοιες που αφορούν αυστηρά το χώρο των Επιστημών, ενώ το STEAM πραγματεύεται τις ίδιες έννοιες με έναν ίσως πιο καλλιτεχνικό-δημιουργικό τρόπο. Το STREAM (Science, Technology, Reading, Engineering, Art, and Math) πηγαίνει ένα βήμα παραπέρα και εισάγει τη συνιστώσα του Read and Write, δηλαδή «διαβάζω, αναπτύσσω κριτική σκέψη και έχω την ικανότητα να αποτυπώσω στο χαρτί τις σκέψεις μου». Υποστηρίζεται ότι δεν αρκούν οι στείρες γνώσεις στα μαθηματικά, τις επιστήμες και την τεχνολογία, αλλά πρέπει να δοθεί ιδιαίτερη έμφαση ώστε τα παιδιά να αναπτύσσουν ικανότητες κριτικής σκέψης για να μπορούν να διαχειριστούν την υπερπληροφόρηση.

Στη σύγχρονη εποχή, η εκπαίδευση STEM έχει γίνει ακόμα πιο σημαντική. Οι τεχνολογικές εξελίξεις έχουν ανοίξει νέους ορίζοντες στην επιστημονική έρευνα και την εφαρμογή. Η ανάγκη για εξειδίκευση σε τομείς όπως η τεχνητή νοημοσύνη, η βιοτεχνολογία και η βιομηχανική αυτοματοποίηση είναι πιο επιτακτική από ποτέ. Η εκπαίδευση STEM δεν απευθύνεται μόνο στην προετοιμασία μελλοντικών επιστημόνων και μηχανικών, αλλά και στην ανάπτυξη δεξιοτήτων που είναι κρίσιμες στη σύγχρονη κοινωνία, όπως η αναλυτική σκέψη, η δημιουργικότητα, η συνεργασία και η επίλυση προβλημάτων. Συνεχίζει να αναπτύσσεται και να προσαρμόζεται στις ανάγκες της σύγχρονης κοινωνίας, προετοιμάζοντας τους νέους για τις προκλήσεις και τις ευκαιρίες που θα δεχθούν.

## ***2.2 Εκπαιδευτική ρομποτική***

Η εκπαιδευτική ρομποτική είναι ένας τομέας της εκπαίδευσης που χρησιμοποιεί τη ρομποτική τεχνολογία ως εργαλείο μάθησης για να διδάξει μαθητές σε διάφορα εκπαιδευτικά επίπεδα. Στην εκπαίδευση STEM η εκπαιδευτική ρομποτική αναδύεται ως ένας σημαντικός κλάδος που ενσωματώνει και μετατρέπει τη θεωρία σε πράξη. Με τη χρήση ρομποτικών

συστημάτων, οι μαθητές δεν περιορίζονται στην απομνημόνευση γνώσεων, αλλά αναπτύσσουν δεξιότητες μέσω του σχεδιασμού και του προγραμματισμού, κατανοώντας πώς οι μαθηματικοί και φυσικοί κανόνες εφαρμόζονται στην πράξη. Έτσι, η εκπαιδευτική ρομποτική δεν είναι απλά μια επιμέρους δραστηριότητα, αλλά ένας εκπαιδευτικός τρόπος που καλλιεργεί δεξιότητες συνεργασίας, επίλυσης προβλημάτων και καινοτομίας. Αποτελεί ουσιαστικά τη «γέφυρα» μεταξύ θεωρίας και πραγματικών εφαρμογών, αναπτύσσοντας στους μαθητές μια βαθύτερη κατανόηση και εμπιστοσύνη στις δυνατότητές τους στο πεδίο του STEM. Ο κύριος στόχος είναι να ενισχύσει την κριτική σκέψη, την επίλυση προβλημάτων, τη συνεργασία και την καινοτομία μέσω της κατασκευής και προγραμματισμού ρομπότ.

Έχουν διεξαχθεί έρευνες σχετικά με τις μεθόδους μάθησης και τον τρόπο με τον οποίο το εκπαιδευτικό περιβάλλον μπορεί να μετατραπεί σε ένα χώρο ενεργού μάθησης και συμμετοχής όταν χρησιμοποιούνται εργαλεία STEM, όπως εκπαιδευτικά ρομπότ. Τα ρομπότ μπορούν να παρέχουν ένα φυσικό περιβάλλον, καθώς προσφέρουν στους μαθητές μια εμπειρία αλληλεπίδρασης με φυσικά αντικείμενα και όχι μόνο με θεωρητικές ή ψηφιακές πληροφορίες, για να επιλύουν προβλήματα μέσω του καινοτόμου παιχνιδιού. Αυτό επιτρέπει στους μαθητές να αντιληφθούν καλύτερα τον τρόπο που λειτουργούν οι μηχανισμοί και οι φυσικές αρχές, καθώς βλέπουν σε πραγματικό χρόνο πώς οι ενέργειές τους επηρεάζουν το ρομπότ. Με τη χρήση αισθητήρων (sensors) και επενεργητών (actuators), τα ρομπότ επιτρέπουν στους μαθητές να εξερευνήσουν και να αλληλοεπιδράσουν με τα πολύπλοκα προβλήματα του πραγματικού κόσμου ενώ προγραμματίζουν [3, 11].

Ως βασικά χαρακτηριστικά της εκπαιδευτικής ρομποτικής αναφέρονται τα εξής:

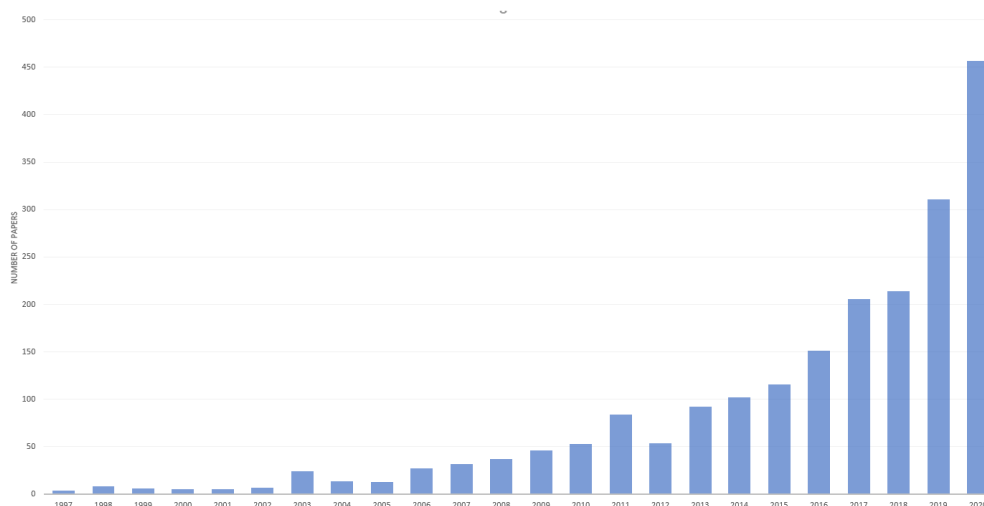
- Διαθεματική μάθηση: Η ρομποτική συνδυάζει διαφορετικά γνωστικά αντικείμενα, όπως μαθηματικά, φυσική, πληροφορική και μηχανολογία. Οι μαθητές καλούνται να εφαρμόσουν γνώσεις από πολλαπλά πεδία για την κατασκευή και τον προγραμματισμό των ρομπότ.
- Πρακτική μάθηση: Η εκπαιδευτική ρομποτική στηρίζεται στην πρακτική εφαρμογή θεωρητικών εννοιών. Οι μαθητές, με τη βοήθεια των ρομπότ, αναπτύσσουν δεξιότητες που συνδέονται με την πραγματική ζωή, π.χ., την κατασκευή μηχανών, την εφαρμογή αλγορίθμων κ.λπ..
- Ομαδική εργασία: Οι μαθητές συνήθως εργάζονται σε ομάδες για την ανάπτυξη των ρομπότ, ενισχύοντας τις δεξιότητες συνεργασίας και επικοινωνίας.
- Επίλυση προβλημάτων: Η εκπαιδευτική ρομποτική επικεντρώνεται στην επίλυση προβλημάτων. Οι μαθητές καλούνται να σχεδιάσουν και να προγραμματίσουν

ρομπότ που εκτελούν συγκεκριμένες λειτουργίες, επιλύοντας τεχνικές και λογικές προκλήσεις.

Αντίστοιχα τα οφέλη της εκπαιδευτικής ρομποτικής είναι τα ακόλουθα:

- Ανάπτυξη κριτικής σκέψης: Οι μαθητές μαθαίνουν να αναλύουν προβλήματα και να σχεδιάζουν λύσεις, ενισχύοντας την αναλυτική τους ικανότητα.
- Τεχνολογικές δεξιότητες: Οι μαθητές αναπτύσσουν βασικές δεξιότητες, μεταξύ άλλων, στον προγραμματισμό, τις τεχνολογίες αιχμής και την μηχανολογία.
- Κίνητρο για μάθηση: Η αλληλεπίδραση με τα ρομπότ και η δυνατότητα να δημιουργούν κάτι χειροπιαστό ενθαρρύνει τους μαθητές να ασχοληθούν περισσότερο με την επιστήμη και την τεχνολογία.
- Προετοιμασία για το μέλλον: Καθώς η ρομποτική και η αυτοματοποίηση γίνονται αναπόσπαστα κομμάτια της βιομηχανίας, οι δεξιότητες που αποκτούν οι μαθητές μέσω της εκπαιδευτικής ρομποτικής θα είναι πολύτιμες σε επαγγελματικά περιβάλλοντα.

Αυτές οι δραστηριότητες βοηθούν στην προετοιμασία των μαθητών για το μελλοντικό τεχνολογικό περιβάλλον και ενθαρρύνουν την ενασχόληση με τα επαγγέλματα STEM. Επιπλέον, η συμμετοχή σε εθνικούς ή παγκόσμιους διαγωνισμούς ρομποτικής δίνει στους μαθητές τη δυνατότητα να ανταγωνιστούν και να συνεργαστούν με συνομηλίκους τους από όλο τον κόσμο. Ωστόσο, η εφαρμογή της εκπαιδευτικής ρομποτικής αντιμετωπίζει προκλήσεις, όπως το υψηλό κόστος εξοπλισμού και η ανάγκη κατάλληλης εκπαίδευσης των εκπαιδευτικών. Παρόλα αυτά, πολλές κυβερνήσεις και οργανισμοί υποστηρίζουν τέτοιες πρωτοβουλίες μέσω χρηματοδοτήσεων και συνεργασιών με την τεχνολογική βιομηχανία. Ταυτόχρονα, αξίζει να αναφερθεί ότι η μείωση του κόστους των εργαλείων που σχετίζονται με τους αυτοματισμούς, τους υπολογιστές και τα εργαλεία ανάπτυξης (χαμηλού κόστους αισθητήρες, ανοικτό λογισμικό, χαμηλού κόστους Η/Υ) σε συνδυασμό με την αυξανόμενη ζήτηση για ψηφιακές δεξιότητες, έχει οδηγήσει σε μια σημαντική άνθηση της εκπαιδευτικής ρομποτικής. Αυτή η ανάπτυξη είναι εμφανής και από την εκθετική αύξηση των δημοσιεύσεων στην επιστημονική κοινότητα, όπως παρουσιάζεται στο [12].



Εικόνα 2-1. Δημοσιεύσεις που αφορούν την εκπαιδευτική ρομποτική ανά έτος [12].

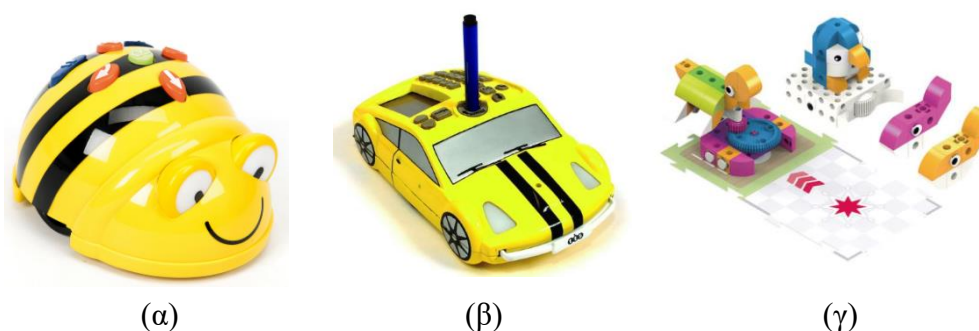
### 2.3 Εκπαιδευτικές πλατφόρμες

Τα τελευταία χρόνια, κατά την αναζήτηση ιδεών για δραστηριότητες STEM, έχουν δημιουργηθεί ενδιαφέρουσες πλατφόρμες που δίνουν έμφαση στην κατανόηση των βασικών εννοιών του STEM. Αυτές οι πλατφόρμες προετοιμάζουν τους μαθητές ώστε να είναι σε θέση να αντιμετωπίσουν και να κατανοήσουν πιο απαιτητικές προκλήσεις καθώς προχωρούν σε μεγαλύτερες τάξεις. Παρακάτω, θα αναφέρουμε ορισμένα παραδείγματα από αυτές. Οι πλατφόρμες διαχωρίζονται με γνώμονα την χρήση τους, την χρηστικότητα και συνδεσμολογία τους, το κόστος τους κ.α.

Ξεκινώντας θα δούμε ορισμένα ρομπότ που έχουν απλούστερη μορφή και απευθύνονται σε μικρότερες ηλικίες, στοχεύοντας σε μια ομαλή εισαγωγή στην έννοια και τη λειτουργία του ρομπότ. Τα προγραμματιζόμενα ρομπότ δαπέδου είναι ειδικά κατασκευασμένα ώστε να χρησιμοποιούνται ακόμη και από μαθητές προσχολικής ηλικίας, αλλά κυρίως των αρχικών τάξεων της δημοτικής εκπαίδευσης. Ο προγραμματισμός τους γίνεται με κουμπιά που βρίσκονται στο σώμα της ρομποτικής συσκευής και μπορούν να προγραμματιστούν για να κινούνται στο χώρο με ακρίβεια όπως κάνει η «χελώνα» της LOGO στον H/Y. Το BeeBot (Εικόνα 2.2 (α)), είναι ένα τέτοιο προγραμματιζόμενο ρομπότ δαπέδου. Η μετεξέλιξη του BeeBot, είναι το ProBot (Εικόνα 2.2 (β)) που έχει τη μορφή ενός αγωνιστικού οχήματος και δίνει τη δυνατότητα στους μαθητές να εξοικειωθούν με τον γραμμικό προγραμματισμό, παρόμοια με το BeeBot. Παράλληλα, τους επιτρέπει να αναβαθμίσουν τις προγραμματιστικές τους δεξιότητες, εισάγοντας μέσω αριθμητικού πληκτρολογίου την απόσταση που πρέπει να διανύσει το όχημα προς μια συγκεκριμένη κατεύθυνση ή τις μοίρες περιστροφής που

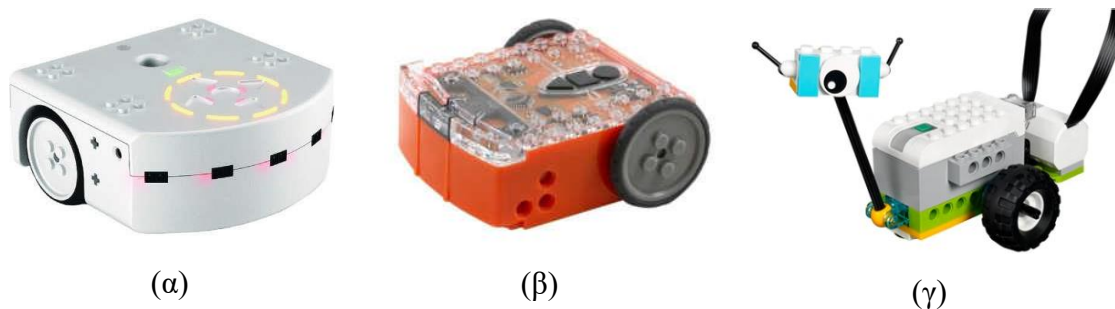


επιθυμούν. Κατά την εκτέλεση των εντολών, αυτές εμφανίζονται στην LCD οθόνη, προσφέροντας στους μαθητές άμεση ανατροφοδότηση και διευκολύνοντας τη διαδικασία μάθησης. Μια αντίστοιχη πλατφόρμα είναι και η Kids First Coding & Robotics (Εικόνα 2.2 (γ)) όπου αποτελείται από ειδικές κάρτες που ορίζουν κατ' ουσίαν τη διαδρομή που ακολουθεί το ρομπότ.



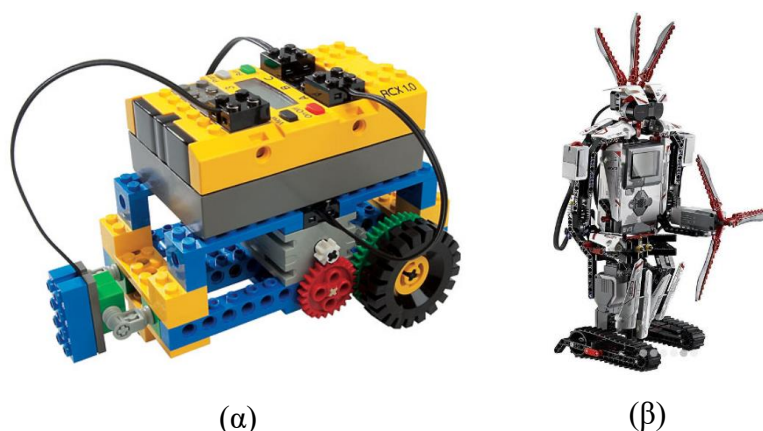
Εικόνα 2-2. Προγραμματιζόμενα ρομπότ δαπέδου, (α) BeeBot, (β) ProBot, (γ) Kids First Coding & Robotics.

Στη συνέχεια, παρουσιάζονται πλατφόρμες που απαιτούν οπτικό προγραμματισμό (visual programming), μια προσέγγιση προγραμματισμού που χρησιμοποιεί οπτικά στοιχεία αντί για γραμμές κώδικα για να δημιουργήσει προγράμματα. Στον οπτικό προγραμματισμό, οι προγραμματιστές χρησιμοποιούν γραφικά αντικείμενα, όπως μπλοκ (block), εικονίδια ή διαγράμματα ροής, για να συνδυάσουν λογικές εντολές και λειτουργίες. Ο εκπαιδευόμενος σε μία προσυναρμολογημένη ρομποτική πλατφόρμα ανάπτυξης (συχνά αυτοκινούμενη) έχει ως κύριο σκοπό να αναπτύξει-βελτιώσει τις προγραμματιστικές του ικανότητες εκπληρώνοντας συγκεκριμένους σκοπούς. Οι πλατφόρμες διαθέτουν αισθητήρες και επενεργητές, οι οποίοι αξιοποιούνται από τον χειριστή-προγραμματιστή για την επίτευξη του σκοπού. Το Thymio II (Εικόνα 2.3 (α)) υποστηρίζει οπτικό προγραμματισμό, επιτρέποντας στους μαθητές να καθορίσουν τις δράσεις και τη συμπεριφορά του μέσω προ-προγραμματισμένων ενεργειών, επιλέγοντας το κατάλληλο πλήκτρο στην επάνω πλευρά της πλατφόρμας. Παρομοίως, το Edison robot (Εικόνα 2.3 (β)) προσφέρει τη δυνατότητα προσαρμογής του ρομποτικού συστήματος, ενώ μπορεί να τροποποιηθεί περαιτέρω με την προσθήκη τουβλακίων Lego, επεκτείνοντας τις δυνατότητές του. Με το LEGO Education WeDo 2.0 (Εικόνα 2.3 (γ)) οι μαθητές κάνουν τα πρώτα τους βήματα στον προγραμματισμό μέσα από μια σειρά διαθεματικών δραστηριοτήτων.



Εικόνα 2-3. Ρομποτικά συστήματα προγραμματιζόμενα, (α) Thymio II, (β) Edison robot, (γ) LEGO Education WeDo 2.0.

Μια άλλη κατηγορία εκπαιδευτικών ρομπότ, αποτελούν αυτά όπου ο εκπαιδευόμενος εκτός από το κομμάτι του προγραμματισμού για την υλοποίηση του σκοπού, πρέπει να συναρμολογήσει-κατασκευάσει κατάλληλους ρομποτικούς μηχανισμούς. Περιλαμβάνουν αισθητήρες, επενεργητές και συνήθως διαθέτουν πλήθος τεχνικών εξαρτημάτων που διευρύνουν τις επιλογές συναρμολόγησης. Ο εκπαιδευόμενος, πέρα από την ενασχόληση με τον προγραμματισμό, χρειάζεται αρχικά να κατασκευάσει και να συναρμολογήσει τον ρομποτικό μηχανισμό. Μία τέτοια πλατφόρμα είναι η Lego Mindstorms RCX (Εικόνα 2.4 (α)) που συνδυάζει προγραμματιζόμενα τούβλα με ηλεκτρικούς κινητήρες, αισθητήρες, κομμάτια Lego και εξειδικευμένα τεχνικά εξαρτήματα (όπως εργαλεία, ακτίνες, άξονες, δοκούς και υδραυλικά μέρη), δίνοντας τη δυνατότητα στον χρήστη να δημιουργήσει ρομπότ ή άλλα αυτοματοποιημένα και διαδραστικά συστήματα. Αντίστοιχη πλατφόρμα αυξημένης πολυπλοκότητας και δυσκολίας είναι η Lego Mindstorms EV3 (Εικόνα 2.4 (β)), ρομπότ που μπορούν να μιλούν, να περπατούν, να σκέφτονται και του οποίου η μορφή δεν έχει περιορισμό και σχηματίζεται από την φαντασία του μαθητή.

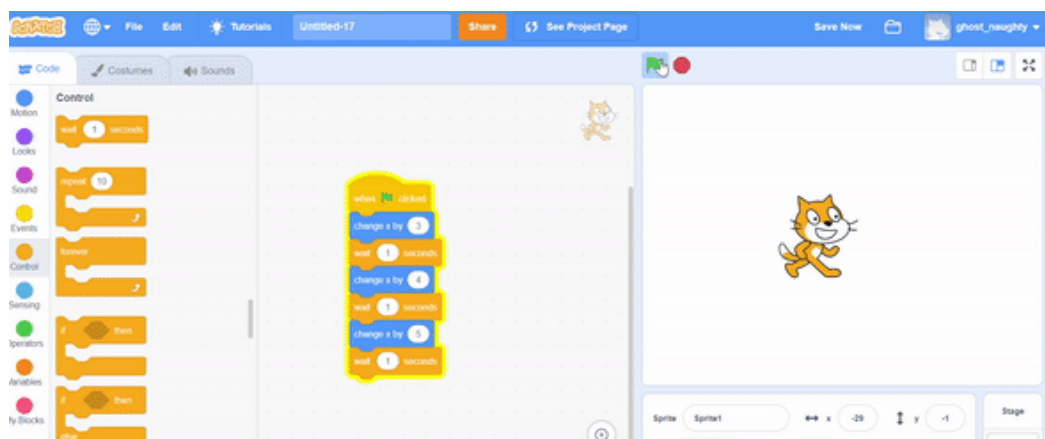


Εικόνα 2-4. Ρομποτικά συστήματα προγραμματιζόμενα και συναρμολογούμενα, (α) Lego Mindstorms RCX, (β) Lego Mindstorms EV3.

## 2.4 Υφιστάμενα λογισμικά εκπαιδευτικής ρομποτικής

### 2.4.1 Scratch

Το Scratch είναι η μεγαλύτερη πλατφόρμα προγραμματισμού για παιδιά παγκοσμίως και προσφέρει μια εύχρηστη γλώσσα προγραμματισμού με οπτικό περιβάλλον (Εικόνα 2.5). Δίνει τη δυνατότητα στους νέους να δημιουργούν ψηφιακές ιστορίες, παιχνίδια και κινούμενα σχέδια. Παράλληλα, ενισχύει την υπολογιστική σκέψη, τις δεξιότητες επίλυσης προβλημάτων, τη δημιουργικότητα στη διδασκαλία και τη μάθηση, την αυτοέκφραση, τη συνεργατική εργασία και την ίση πρόσβαση στην πληροφορική. Έχει αναπτυχθεί από μία ομάδα ερευνητών του MIT (Lifelong Kindergarten Group, MIT Media Lab) το 2003 [13]. Έχει σχεδιαστεί κυρίως για παιδιά ηλικίας 8-16, αλλά απευθύνεται σε ανθρώπους όλων των ηλικιών.



Εικόνα 2-5. Περιβάλλον Scratch.

Για τα παιδιά μικρότερης ηλικίας (5-7) υπάρχει μία απλοποιημένη έκδοση, το ScratchJr (Εικόνα 2.6). Ο προγραμματισμός με το Scratch θυμίζει σε μεγάλο βαθμό τη δημιουργία ενός θεατρικού σκηνικού από την αρχή. Υπάρχει μια κενή σκηνή (η οθόνη της εφαρμογής), όπου ο χρήστης μπορεί να επιλέξει έτοιμες εικόνες ή να δημιουργήσει τις δικές του, και στη συνέχεια να τοποθετήσει αντικείμενα της επιλογής του. Αυτά τα αντικείμενα μπορούν να εκτελούν διάφορες κινήσεις και να αλληλεπιδρούν μεταξύ τους. Ένας βασικός σχεδιαστικός στόχος του Scratch είναι να υποστηρίξει την αυτοκατευθυνόμενη μάθηση μέσω της επεξεργασίας και της συνεργασίας με συνομηλίκους [14].



Εικόνα 2-6. Περιβάλλον ScratchJr.

Μια επιπλέον έκδοση του Scratch είναι το S4A (Scratch for Arduino). Το S4A είναι μια τροποποίηση που παρέχει την δυνατότητα απλού προγραμματισμού της πλατφόρμας Arduino (Εικόνα 2.7). Προσφέρει νέα μπλοκ ώστε να διαχειρίζονται οι αισθητήρες και επενεργητές που είναι συνδεδεμένοι στο Arduino. Διατίθεται επίσης μια πλακέτα αναφοράς αισθητήρων συναφή με αυτή του PicoBoard. Απευθύνεται κυρίως στην ηλικιακή ομάδα των 10-12 ετών. Στα πλεονεκτήματα του λογισμικού είναι η συμβατότητα με το Scratch, μπορεί να διαβάσει projects που δημιουργήθηκαν με το Scratch, όχι όμως το αντίθετο.



Εικόνα 2-7. Περιβάλλον Snap4Arduino.

Στα μειονεκτήματα του είναι ότι υπάρχουν σημαντικοί περιορισμοί στην εκμετάλλευση των πόρων του μικροελεγκτή. Συγκεκριμένα, υπάρχει ικανοποιητική εκμετάλλευση

αναλογικών θυρών για ανάγνωση έξι αισθητήρων, αλλά μόνο τριών ψηφιακών και τριών ψευδοαναλογικών εξόδων για έλεγχο επενεργητών και λοιπών διατάξεων εξόδου. Επιπλέον ο μικροελεγκτής θα πρέπει να έχει προφορτωμένο συγκεκριμένο firmware, για να μπορεί να επιτυγχάνεται επικοινωνία. Τέλος, αξίζει ιδιαίτερη προσοχή ότι το Arduino ανταλλάσσει πληροφορίες με το λογισμικό με αργό ρυθμό, κάθε 75msec [15].

Το Scratch είναι ένα εργαλείο οπτικής γλώσσας προγραμματισμού υψηλού επιπέδου που αλληλοεπιδρά με τους χρήστες μέσω διαγραμμάτων και μπλοκ που έχουν ενσωματωμένα τα βασικά ενός προγράμματος. Το Scratch χρησιμοποιείται για τη δημιουργία διαδραστικών προγραμμάτων ειδικά για παιδιά που χρησιμοποιούν τις διεπαφές μπλοκ, έτσι ώστε να μπορούν να μαθαίνουν εύκολα γλώσσες και είναι δωρεάν στη χρήση.

#### 2.4.2 Snap!

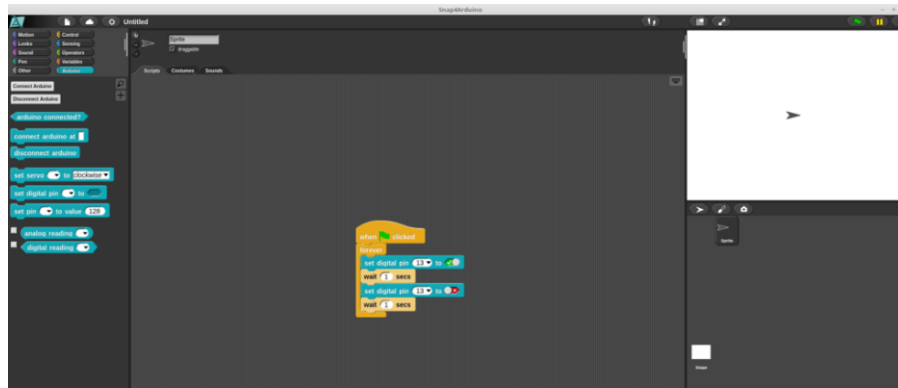
Το Snap! παρουσιάστηκε από το Πανεπιστήμιο της Καλιφόρνια στο Μπέρκλεϋ και είναι μια οπτική γλώσσα προγραμματισμού. Είναι μια επανασχεδίαση του λογισμικού Scratch που επιτρέπει στον χρήστη να δημιουργεί και να προσθέτει τα δικά του μπλοκ στα προγράμματα που δημιουργεί. Οι δημιουργοί του λογισμικού στοχεύουν στη δημιουργία μίας γλώσσας οπτικού προγραμματισμού για διδασκαλία πληροφορικής (όπως και το Scratch), στοχεύοντας σε μαθητές γυμνασίου ή πανεπιστημίου. Ο χρήστης μπορεί να δημιουργήσει νέες δομές ελέγχου, όπως έναν βρόχο for, γράφοντας ένα σενάριο (Εικόνα 2.8) όπως φαίνεται στα αριστερά. Μόλις δημιουργηθεί το μπλοκ for, μπορεί να χρησιμοποιηθεί ακόμη και για τη δημιουργία ένθετων βρόχων, όπως φαίνεται στα δεξιά [16].



Εικόνα 2-8. Παράδειγμα λογισμικού Snap!.

Το Snap4Arduino (Snap! for Arduino) είναι μία τροποποίηση του Snap! για τον προγραμματισμό της πλατφόρμας Arduino (Εικόνα 2.9), από τους δημιουργούς του S4A. Το λογισμικό λειτουργεί με την ίδια ακριβώς λογική, όπως και το S4A, με τις όποιες τροποποιήσεις-βελτιώσεις περιλαμβάνει το Snap! σε σχέση με το Scratch.

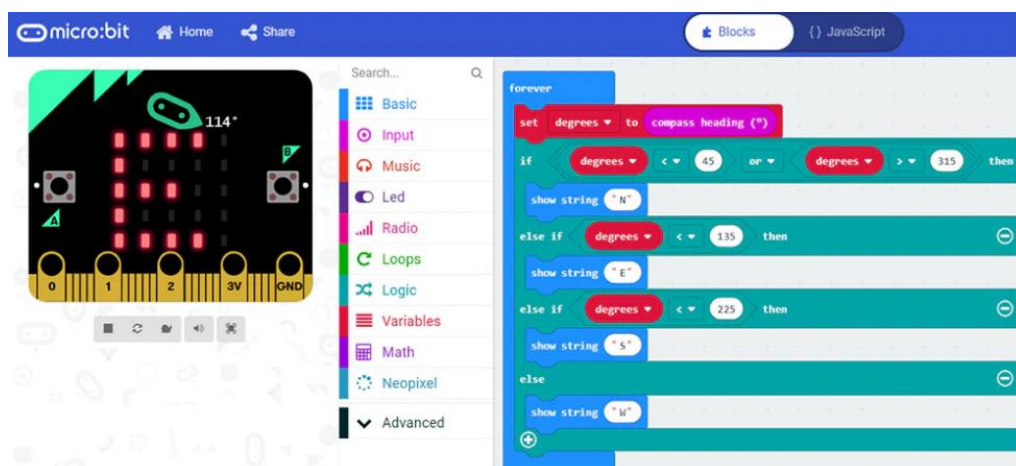




Εικόνα 2-9. Περιβάλλον Snap4Arduino.

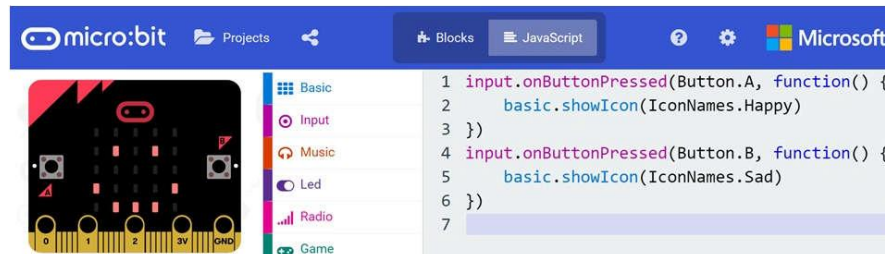
### 2.4.3 MakeCode

Το MakeCode είναι μια πλατφόρμα ανάπτυξης λογισμικού που δημιουργήθηκε από τη Microsoft και στοχεύει να βοηθήσει τους εκπαιδευτικούς, τους μαθητές και τους χομπίστες να μάθουν προγραμματισμό με εύκολο και διαδραστικό τρόπο. Το MakeCode υποστηρίζει διάφορες συσκευές και περιβάλλοντα, παρέχοντας ένα οπτικό, block-based σύστημα προγραμματισμού, που μοιάζει με παζλ, αλλά και τη δυνατότητα χρήσης πραγματικού κώδικα σε γλώσσες όπως η JavaScript και η Python. Η πλατφόρμα χρησιμοποιείται σε εκπαιδευτικά πλαίσια για να κάνει τον προγραμματισμό πιο προσιτό και διασκεδαστικό, και βοηθά τους χρήστες να αναπτύξουν δεξιότητες επίλυσης προβλημάτων και λογικής. Διατίθεται σε πολλές εκδόσεις με δυνατότητες ανάπτυξης κώδικα για διάφορες δημοφιλείς εκπαιδευτικές πλατφόρμες υλικού (BBC: micro:bit, Lego EV3 κ.ά.) και λογισμικού (Minecraft) που κυκλοφορούν (Εικόνα 2.10).



Εικόνα 2-10. Περιβάλλον MakeCode for micro:bit.

Το λογισμικό MakeCode δεν χρειάζεται κάποιας μορφής εγκατάσταση (τρέχει σε όλους τους σύγχρονους browsers). Διατίθεται προσομοιωτής των φυσικών συσκευών και άρα μπορεί ο χρήστης να δοκιμάζει την ορθότητα του κώδικα (Εικόνα 2.11) και χωρίς την ύπαρξη της πραγματικής συσκευής.

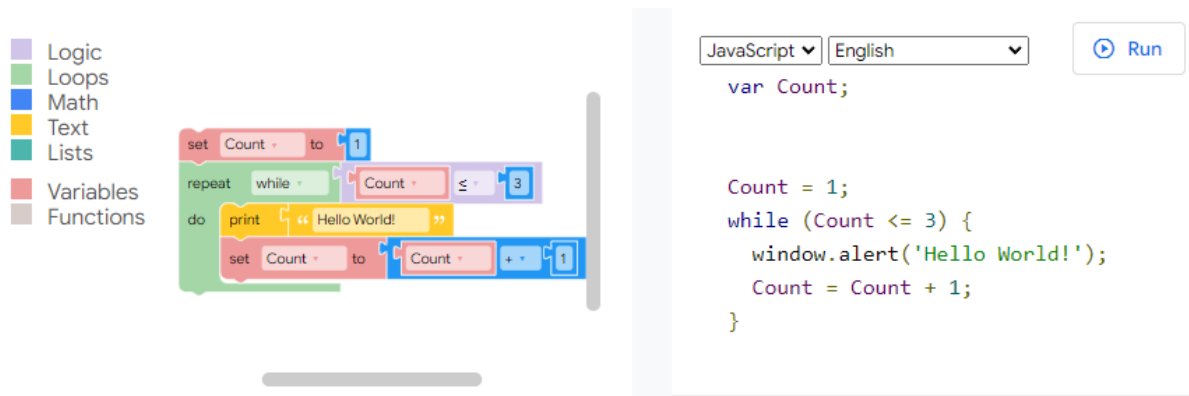


Εικόνα 2-11. Περιβάλλον MakeCode for micro:bit (JavaScript).

#### 2.4.4 Blockly

Το Blockly είναι μια ανοικτού κώδικα βιβλιοθήκη, που αναπτύχθηκε από την Google, επιτρέποντας στους χρήστες να προγραμματίζουν χρησιμοποιώντας οπτικά μπλοκ αντί για κείμενο [17]. Τα μπλοκ στο Blockly συνδέονται μεταξύ τους σαν κομμάτια παζλ, αντιπροσωπεύοντας έννοιες προγραμματισμού όπως μεταβλητές, βρόχους, συνθήκες κ.λπ. (Εικόνα 2.12). Ο στόχος του Blockly είναι να διευκολύνει την κατανόηση του προγραμματισμού, ιδιαίτερα για αρχάριους και παιδιά. Παρόμοια με το MakeCode, το Blockly παρέχει έναν οπτικό τρόπο προγραμματισμού με μπλοκ που μπορούν να συρθούν και να συνδεθούν, κάνοντας τον προγραμματισμό πιο προσβάσιμο και κατανοητό. Αν και οι χρήστες δουλεύουν με μπλοκ, το Blockly μπορεί να μετατρέψει αυτά τα μπλοκ σε κώδικα σε διάφορες γλώσσες προγραμματισμού, όπως JavaScript, Python, Lua, PHP, Dart. Είναι μια βιβλιοθήκη που μπορεί να ενσωματωθεί σε άλλες πλατφόρμες. Για παράδειγμα, πολλές εκπαιδευτικές πλατφόρμες, όπως το Scratch, χρησιμοποιούν τεχνολογίες και ιδέες που σχετίζονται με το Blockly. Επιτρέπει στους χρήστες να πειραματίζονται με τη δομή του κώδικα, δοκιμάζοντας νέες ιδέες και αλλάζοντας την αλληλουχία των μπλοκ με εύκολο τρόπο.

Το Blockly χρησιμοποιείται ευρέως στην εκπαίδευση και είναι η βάση για πολλά προγράμματα που στοχεύουν να διδάξουν προγραμματισμό με τρόπο που δεν απαιτεί προηγούμενη γνώση γλωσσών προγραμματισμού. Συνήθως χρησιμοποιείται σε πλατφόρμες και εφαρμογές που έχουν ως στόχο να βοηθήσουν μαθητές και αρχάριους να μάθουν προγραμματισμό με παιχνιδιάρικο και διαισθητικό τρόπο.



Εικόνα 2-12. Περιβάλλον Google Blockly.

Λόγω των εξαιρετικών δυνατοτήτων του, αποτέλεσε το βασικό δομικό στοιχείο με το οποίο αναπτύχθηκε το λογισμικό στην παρούσα εργασία. Ο τρόπος λειτουργίας του καθώς και οι δυνατότητές του αναλύονται διεξοδικά σε επόμενο κεφάλαιο.

## 2.5 Προκλήσεις

Κατά τη φάση του σχεδιασμού και της αρχικής ανάπτυξης, δόθηκε ιδιαίτερη έμφαση στη φιλικότητα προς τον χρήστη, καθώς αυτό είχε αναγνωριστεί ως ένα από τα κύρια προβλήματα στις υπάρχουσες πλατφόρμες. Πολλές από αυτές τις πλατφόρμες δεν ήταν εύχρηστες, γεγονός που περιόριζε την αποτελεσματική αλληλεπίδραση των μαθητών με το σύστημα. Η ευκολία χρήσης θεωρήθηκε κρίσιμη προϋπόθεση για να κερδίσει και να διατηρήσει το ενδιαφέρον των μαθητών, ειδικά στις ηλικίες της Πρωτοβάθμιας και Δευτεροβάθμιας εκπαίδευσης.

Παράλληλα, αναγνωρίστηκε η ανάγκη για επεκτασιμότητα του συστήματος, ώστε να υπάρχει η ευελιξία για την πραγματοποίηση ποικίλων εκπαιδευτικών σεναρίων. Η δυνατότητα εμπλουτισμού του συστήματος με νέα τμήματα επιτρέπει τη συνεχή προσαρμογή του εργαλείου στις ανάγκες του εκπαιδευτικού περιβάλλοντος και την εξέλιξη των μαθητών, ενισχύοντας έτσι τη δημιουργική τους σκέψη και την ικανότητα επίλυσης προβλημάτων.

Ιδιαίτερη σημασία απαίτησε η μείωση της πολυπλοκότητας του συστήματος, τόσο σε επίπεδο hardware όσο και software, ώστε να εξασφαλιστεί ότι το σύστημα θα είναι προσιτό και κατανοητό για μαθητές με μικρή εμπειρία στον προγραμματισμό ή τη ρομποτική. Στόχος ήταν να ελαχιστοποιηθεί η καμπύλη εκμάθησης και να αποφευχθεί η αποθάρρυνση των μαθητών λόγω περίπλοκων διαδικασιών.

Επιπλέον, δόθηκε προτεραιότητα στη διατήρηση χαμηλού κόστους, χρησιμοποιώντας ανοικτά εργαλεία ανάπτυξης. Αυτό επιτρέπει σε περισσότερα σχολεία και εκπαιδευτικούς



φορείς να υιοθετήσουν την πλατφόρμα, καθιστώντας την προσιτή σε μεγαλύτερο κοινό, χωρίς να θυσιάζονται οι βασικές της λειτουργίες ή η ποιότητα της εκπαιδευτικής εμπειρίας.

## ΚΕΦΑΛΑΙΟ 3: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ HARDWARE ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

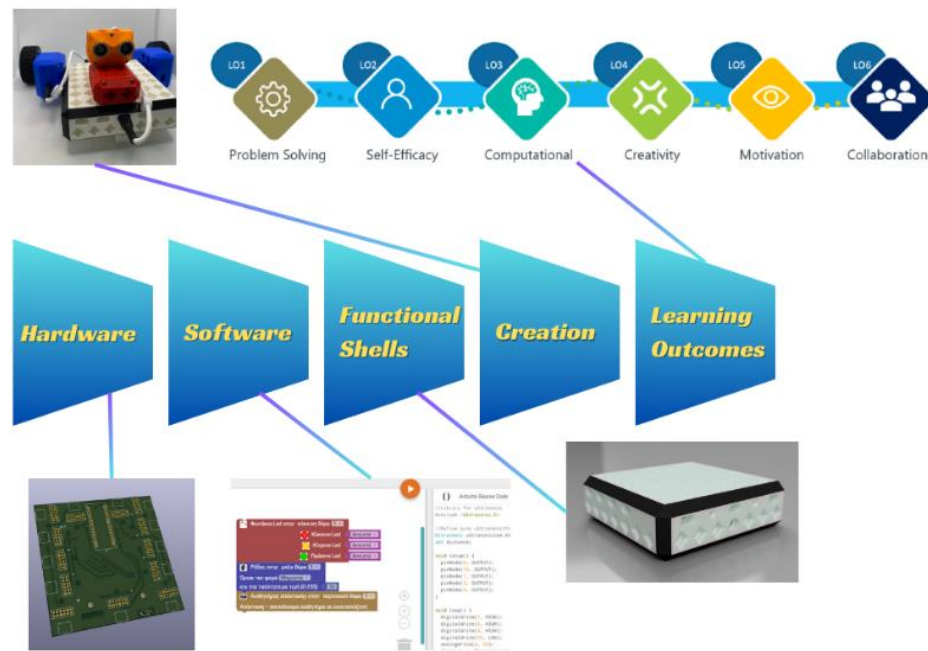
---

Στο κεφάλαιο αυτό παρουσιάζεται η σχεδιαστική φιλοσοφία με βάση την οποία πραγματοποιήθηκε η ανάπτυξη του υλικού (hardware) του συστήματος ColourBot. Επιπλέον παρουσιάζονται λεπτομερώς τα σχέδια που αναπτύχθηκαν, τα ηλεκτρονικά που επιλέχθηκαν και τα πρωτότυπα που δημιουργήθηκαν. Ο σχεδιασμός είναι απλός αλλά και αποτελεσματικός. Η ανάπτυξη του γίνεται πλήρως με χρήση open source λογισμικών και εργαλείων όπως το ‘Kicad for PCB design’. Τα ηλεκτρονικά εξαρτήματα του μπορούν να κατανεμηθούν σε τρεις κατηγορίες, την *κεντρική μονάδα (Mainboard)*, τους *αισθητήρες και επεξεργαστές* και τη *μονάδα ισχύος (Power unit)* όπου θα αναλυθούν παρακάτω.

### 3.1 Σχεδιαστική φιλοσοφία

Ο βασικός στόχος αυτής της εργασίας ήταν η ανάπτυξη μιας εκπαιδευτικής πλατφόρμας που θα ανταποκρινόταν σε διάφορες σχεδιαστικές και λειτουργικές απαιτήσεις, ενώ ταυτόχρονα θα λάμβανε υπόψη συγκεκριμένους εκπαιδευτικούς στόχους και θα ήταν εύκολα διαχειρίσιμη από τον χρήστη.

Η σχεδιαστική φιλοσοφία, έχει ως στόχο να προσφέρει ένα επεκτάσιμο, αρθρωτό σύστημα που χαρακτηρίζεται από χαμηλή πολυπλοκότητα. Αυτός ο σχεδιασμός είναι κατάλληλος για μαθητές με ελάχιστη έως καθόλου γνώση σε σχετιζόμενα θέματα. Ο κύκλος ανάπτυξης του συστήματος παρουσιάζεται στην Εικόνα 3.1.



Εικόνα 3-1. Κύκλος ανάπτυξης.

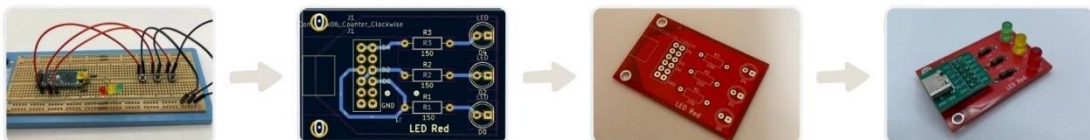
Η σχεδιαστική προσέγγιση έχει επικεντρωθεί, όπως αναφέρθηκε προηγουμένως, στην επεκτασιμότητα, η οποία επιτρέπει την ανάπτυξη και ενίσχυση του συστήματος με νέα τμήματα (modules) όποτε αυτό χρειαστεί. Επίσης, η πολυπλοκότητα του συστήματος παρουσιάζεται με τρόπο που επιτρέπει στους μαθητές να συνθέτουν διάφορα σενάρια, συναρμολογώντας τα διαθέσιμα μέρη. Έτσι, ενισχύεται η δημιουργική τους σκέψη χωρίς να προκαλείται άγχος ή αποθάρρυνση. Για να επιτευχθούν αυτοί οι στόχοι, είναι σημαντικό ο μαθητής να διαθέτει το ελάχιστο δυνατό επίπεδο εκνευρισμού, καθώς υψηλή δυσκολία στη χρήση ή/και στο σχεδιασμό και την υλοποίηση μπορεί να τον αποθαρρύνει.

Με βάση τα παραπάνω και λαμβάνοντας υπόψιν τα διαθέσιμα εργαλεία, σχεδιάστηκε ένα σύστημα που είχε σαν πυρήνα ένα μικροελεγκτή από την οικογένεια Arduino. Έχουν σχεδιαστεί και παραχθεί απλές ηλεκτρονικές συσκευές με ενσωματωμένους αισθητήρες και επενεργητές που συνδέονται με μία μοναδική μονάδα ελέγχου, μεγιστοποιώντας έτσι την απόδοση και ελαχιστοποιώντας το κόστος. Όλες οι συσκευές προγραμματίζονται και ελέγχονται μέσω μιας προσαρμοσμένης γραφικής διεπαφής που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας και θα αναλυθεί με λεπτομέρεια στο Κεφάλαιο 5. Οι ηλεκτρονικές συσκευές όσο και η κύρια μονάδα ελέγχου, περικλείονται από προσαρμοσμένα λειτουργικά κελύφη τα οποία επιτρέπουν στους μαθητές να τα χρησιμοποιούν ως δομικά στοιχεία για τις λειτουργικές τους συσκευές. Αυτή η προσέγγιση συνδέεται με το πόρισμα του [18] λαμβάνοντας υπόψη τα κριτήρια για την επιλογή συσκευών που σχετίζονται με το STEM.

Όπως αναφέρθηκε ήδη, η προτεινόμενη προσέγγιση βασίζεται στις εργασίες που παρουσιάζονται στα [4, 5] ωστόσο διαφέρει σημαντικά. Όσον αφορά τα ηλεκτρονικά, υπήρξε ένας πλήρης επανασχεδιασμός εστιάζοντας σε έναν πιο ευέλικτο και καθολικό μηχανισμό για τη σύνδεση του υλικού υιοθετώντας τον συνδετήρα τύπου C καθώς θεωρήθηκε πιο πρακτικός, πιο ευέλικτος και δεν έχει κατεύθυνση σε σύγκριση με την πρωτότυπη προσέγγιση που βασίζεται στο συνδετήρα RJ45.

Όλα τα ηλεκτρονικά επανασχεδιάστηκαν εστιάζοντας σε μειωμένο μέγεθος και χαμηλότερο κόστος. Ο επεξεργαστής που υιοθετήθηκε για τη μονάδα ελέγχου είναι ο Arduino Nano, αντικαθιστώντας το Arduino Mega 2560, λαμβάνοντας επομένως λιγότερο χώρο στην πλακέτα και ταυτόχρονα μειώνοντας την κατανάλωση ρεύματος. Όλα τα ηλεκτρονικά λειτουργούν στην λογική των 5V.

Κατά την κατασκευή της πλακέτας, αρχικά πραγματοποιήθηκαν δοκιμές σε πλακέτες δοκιμών (breadboard) για τη διασφάλιση της σωστής συνδεσμολογίας των εξαρτημάτων και την επιβεβαίωση της ορθής λειτουργίας κάθε κυκλώματος. Στη συνέχεια, σχεδιάστηκαν τα σχηματικά και οι διατάξεις PCB μέσω εξειδικευμένου λογισμικού και ακολούθησε η αποστολή τους σε εξωτερικό κατασκευαστικό κέντρο για την υλοποίηση των τελικών πλακετών. Τέλος, τα ηλεκτρονικά εξαρτήματα συγκολλήθηκαν προσεκτικά στις ολοκληρωμένες πλακέτες, ολοκληρώνοντας τη διαδικασία κατασκευής (Εικόνα 3.2).



Εικόνα 3-2. Διαδικασία κατασκευής πλακέτας.

Το προτεινόμενο σύστημα αποτελείται από διάφορα αρθρώματα: έναν άρθρωμα για την τροφοδοσία, την κεντρική πλακέτα στην οποία συνδέεται ο μικροελεγκτής και τους διάφορους αισθητήρες και επενεργητές. Η πολυπλοκότητα του συστήματος καθορίζεται από τις επιλογές του χρήστη και τον τρόπο με τον οποίο μπορεί να συνδέσει τα επιλεγμένα αρθρώματα στην κεντρική μονάδα και ως εκ τούτου στον μικροελεγκτή Arduino. Η εκτέλεση του αυτοματισμού που επιθυμούμε πραγματοποιείται μέσω του λογισμικού που είναι εγκατεστημένο στον υπολογιστή. Ο χρήστης εισάγει τον κώδικα χρησιμοποιώντας μπλοκ, όπως θα εξηγηθεί αναλυτικά παρακάτω.

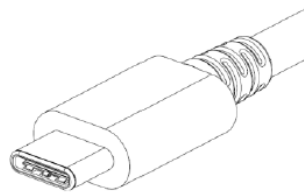
Τέλος, το σύστημα διαθέτει 12 εισόδους/εξόδους για την κεντρική πλακέτα. Αυτές οι είσοδοι/εξοδοι μπορούν να διαχωριστούν ανάλογα με τη λειτουργικότητά τους (Πίνακας 3-1). Κάθε μία από αυτές έχουν ένα χρώμα και έναν αριθμό που επιτρέπει στον μαθητή να αναγνωρίσει εύκολα πού να συνδέσει κάθε συσκευή. Αυτή η έγχρωμη προσέγγιση ελαχιστοποιεί τον χρόνο συναρμολόγησης και αφαιρεί ένα επίπεδο απογοήτευσης από τους μαθητές, επειδή δεν χρειάζεται να ασχοληθούν με περίπλοκες συνδέσεις και ενδεχόμενο εκνευρισμό που μπορεί να έχει αποτρεπτικό χαρακτήρα για μικρά παιδιά.

<b>Πίνακας 3-1. Χρωματικός κώδικας συνδέσεων.</b>	
<b>Χρώμα</b>	<b>Σήματα</b>
Κόκκινο	3 Digital
Πορτοκαλί	2 Analog
Μπλε	1 Digital & 1 PWM
Πράσινο	1 Digital / Analog
Κίτρινο	1 PWM
Ροζ	2 Analog

## 3.2 Σύνδεση δομικών στοιχείων

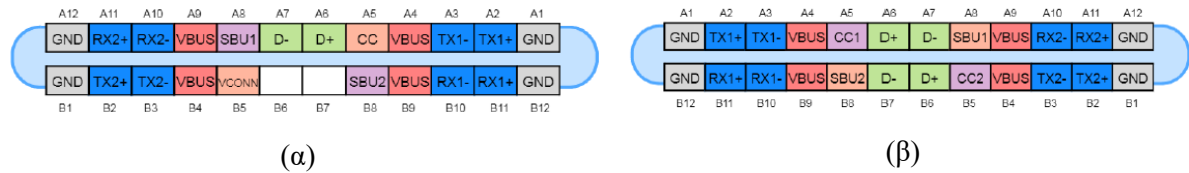
### 3.2.1 Γενικές πληροφορίες των USB Type-C

Όλα τα σχέδια έχουν ως κοινό στοιχείο τον συνδετήρα Type-C ώστε να εξασφαλίζεται η τροφοδοσία τους και η επικοινωνία με την κεντρική πλακέτα. Το καλώδιο USB Type-C είναι σε θέση να γίνει το «παγκόσμια» καλώδιο, καθώς είναι ικανό να παρέχει γρήγορες ταχύτητες μεταφοράς δεδομένων έως και 10Gb/s, 100W συνεχούς ροής ισχύος και δυνατότητες βίντεο εξαιρετικά υψηλού εύρους ζώνης που διατίθενται μέσω των εναλλακτικών λειτουργιών όλων παράλληλα με ενιαία σύνδεση.



Εικόνα 3-3. Καλώδιο USB Type-C.

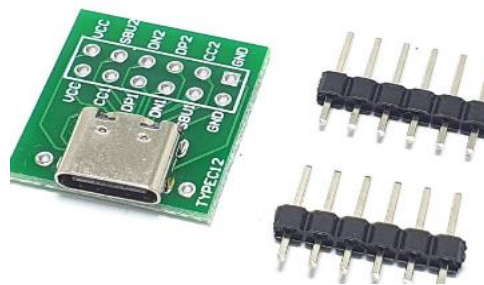
Ο συνδετήρας έχει 24 ακροδέκτες. Εξαιτίας της αναστρεψιμότητας τους, οι ακροδέκτες είναι διατεταγμένοι με κατοπτρική διαμόρφωση. Υπάρχουν στο σύνολο έξι διαφορετικά ζευγάρια σε πλήρως λειτουργική συγκρότηση και τέσσερις ακροδέκτες όπου προσφέρουν νέες λειτουργίες στο USB (CC1, CC2, SBUS1, SBUS2). Ο σχεδιασμός του συνδετήρα επιτρέπει ταυτόχρονη μεταφορά δεδομένων και παροχή ισχύος. Οι συνδέσεις του USB Type-C είναι πολύ πιο ευέλικτες, οι θύρες μπορεί να έχουν ως λειτουργία την υποδοχή μόνο, την συσκευή μόνο, ή διπλού ρόλου και οι ρόλοι δεδομένων και τροφοδοσίας μπορούν να είναι ανεξάρτητοι. Η εναλλαγή γίνεται δυναμικά με την χρήση πρωτοκόλλου USB Power Delivery. Το πρωτόκολλο USB Power Delivery είναι ένα πρωτόκολλο μονού άκρου, ενός καλωδίου που δημιουργήθηκε από το USB-IF και καθορίζει τις μεθόδους για σειριακή επικοινωνία μέσω του καλωδίου USB Type-C CC. Μια απλή εφαρμογή τους μπορεί να είναι επίσης αποδοτική. Οι θύρες του δεν απαιτείται να υλοποιούν όλα τα προηγμένα χαρακτηριστικά που τις χαρακτηρίζουν. Το ελάχιστο σετ χαρακτηριστικών τους περιλαμβάνει την σύνδεση USB2.0, την ανίχνευση σύνδεσης και αποσύνδεσης καλωδίου και τέλος την ενεργή τροφοδοσία VCONN. Όσο αναφορά τις εναλλακτικές λειτουργίες, το καλώδιο επιτρέπει την χρήση οποιουδήποτε τρίτου πρωτοκόλλου εφόσον το καλώδιο μπορεί να το υποστηρίξει. Εναλλακτικές λειτουργίες διαπραγματεύονται και εισάγονται από θύρα σε θύρα χρησιμοποιώντας το πρωτόκολλο USB Power Delivery. Τα ακόλουθα σήματα ενδέχεται να εκχωρηθούν εκ νέου κατά την είσοδο σε μια εναλλακτική λειτουργία: TX1+/-, RX1+/-, TX2+/-, RX2+/-, SBU1/SBU2. Ξεχωριστές προδιαγραφές ορίζουν τους κανόνες για κάθε εναλλακτική λειτουργία. Επί του παρόντος, υπάρχουν προδιαγραφές για το DisplayPort (VESA) και ThunderBolt (Intel). Οι βασικές δυνατότητες ρεύματος είναι 1.5A@5V, 3A@5V. Είναι σχεδιασμένα να υποστηρίζουν τις τρέχουσες γενιές USB2.0(480 Mb/s), USB3.0(5Gb/s), USB3.1(10Gb/s), και μελλοντικές προδιαγραφές που θα φτάνουν έως και 20Gb/s data rates. Έχει δυο σετ ζευγαριών USB2.0 στον συνδετήρα (connector) όπου συνδέονται σε ένα μοναδικό διαφορετικό ζευγάρι, τα pins D+, D-. Για την επικοινωνία USB3.0 μόνο το ένα από το σετ των TX/RX διαφορετικών ζευγαριών μπορεί να χρησιμοποιηθεί με την χρήση πολυπλέκτη.



Εικόνα 3-4. Συνδετήρας (connector) USB Type-C, (α) Βύσμα USB Type-C, (β) Υποδοχή USB Type C [19].

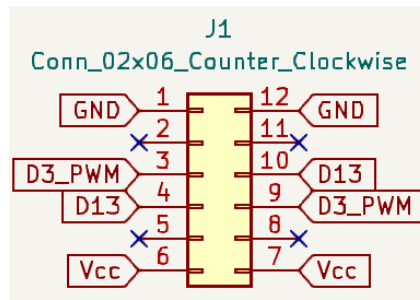
### 3.2.2 Εφαρμογή της Type-C συνδεσμολογίας

Η επιλογή του USB Type-C έγινε κυρίως λόγω της ευελιξίας που προσφέρει. Έχοντας κατοπτρική διάταξη εσωτερικά στα pins δεν προκαθορίζεται η φορά σύνδεσης του. Μπορεί να συνδεθεί το καλώδιο με την υποδοχή και από τις δύο πλευρές του βύσματος. Αυτό αφού απευθυνόμαστε κυρίως σε χρήστες μικρής ηλικίας αυτό μειώνει την πολυπλοκότητα. Όλα τα αρθρώματα επικοινωνούν μεταξύ τους μέσω του συγκεκριμένου πρωτοκόλλου. Τα καλώδια που χρησιμοποιήθηκαν είναι τα τυποποιημένα εργοστασιακά, ενώ στις πλακέτες χρησιμοποιήθηκε ως υποδοχή ένα USB Type-C Breakout Board όπως απεικονίζεται στην Εικόνα 3.5.



Εικόνα 3-5. USB Type-C Breakout Board - 12 Pin Header.

Η πλακέτα αυτή δίνει πρόσβαση σε 12 κεφαλές, δηλαδή έξι ζεύγη pins VCC/VCC, CC1/CC2, DP1/DP2, DN1/DN2, SBU1/SBU2, GND/GND. Τα VCC και GND είναι ήδη βραχυκυκλωμένα. Σε όλα τα σχέδια βραχυκυκλώθηκαν και τα DP1/DP2, DN1/DN2 ώστε να μεταφέρονται αναλογικά ή ψηφιακά σήματα. Χρησιμοποιήθηκε το VCC για όλες τις τροφοδοσίες από την κεντρική πλακέτα προς τις υπόλοιπες και το GND για την γείωση. Τα άλλα δύο δεν χρησιμοποιήθηκαν. Ένα ενδεικτικό σχηματικό διάγραμμα του Breakout Board παρουσιάζεται στην Εικόνα 3.6:

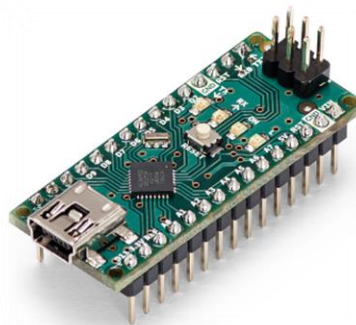


Εικόνα 3-6. Connector Schematic.

### 3.3 Κεντρική Μονάδα Συστήματος

Η κεντρική μονάδα του συστήματος, είναι η μονάδα ελέγχου και αποτελείται από έναν μικροελεγκτή, δώδεκα υποδοχές Type-C και μία υποδοχή με δυο ακροδέκτες για τροφοδοσία. Η επιλογή των εξαρτημάτων έγινε με κριτήριο το χαμηλό κόστος, την υψηλή απόδοση και την χαμηλή κατανάλωση ενέργειας, κάνοντας τον σχεδιασμό ιδανικό για φορητή χρήση και κατάλληλο για τροφοδοσία ρεύματος με μπαταρίες. Ο σχεδιασμός είναι οργανωμένος για να μεγιστοποιεί την αξιοποίηση του χώρου, εξασφαλίζοντας ένα μικρό PCB χωρίς λειτουργικούς συμβιβασμούς.

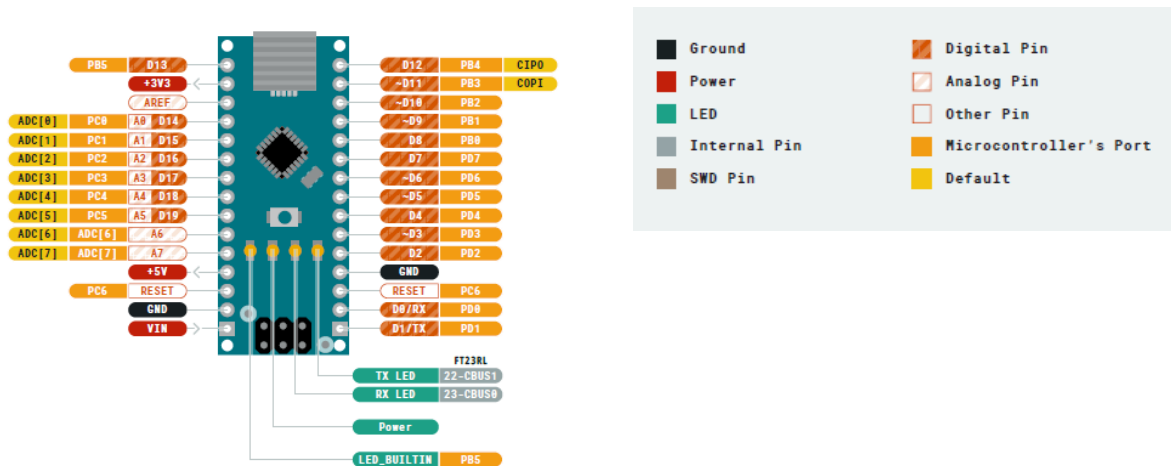
Ο μικροελεγκτής κατέχει κύριο ρόλο καθώς λειτουργεί ως εγkéφαλος του συστήματος καθώς επεξεργάζεται και εκτελεί εντολές για τον έλεγχο όλων των λειτουργιών. Δέχεται δεδομένα από τον χρήστη μέσω του προγραμματιστικού περιβάλλοντος και παράγει τα αντίστοιχα αποτελέσματα. Οι πληροφορίες αυτές διαβιβάζονται και κατανέμονται στα διάφορα εξαρτήματα μέσω των ακροδεκτών. Επιλέχθηκε ο μικροελεγκτής Arduino Nano (5 V power) (Εικόνα 3.7) όπου είναι η κλασική πλακέτα της Arduino με τις μικρότερες διαστάσεις.



Εικόνα 3-7. Arduino Nano.

Αποτελείται από τον επεξεργαστή ATmega328, διαθέτει 14 ψηφιακά και 8 αναλογικά pins, ενώ 6 από τα ψηφιακά υποστηρίζουν και Pulse Width Modulation (PWM) σήματα.



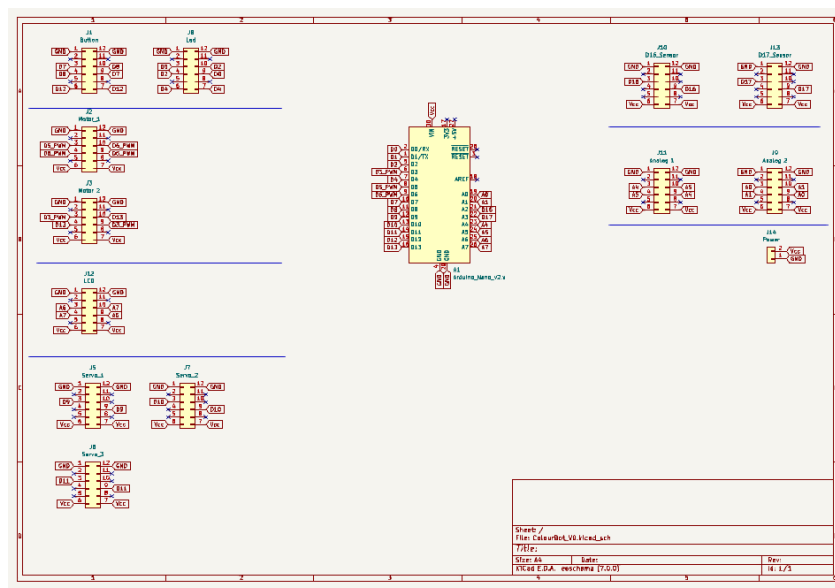


Εικόνα 3-8. Arduino Nano Pinout [20].

Οι ακροδέκτες του μικροελεγκτή, στην περίπτωση μας της πλακέτας Arduino, μπορούν να χωριστούν σε τρεις κύριες κατηγορίες: (i) Οι ψηφιακοί (Digital) ακροδέκτες, που λειτουργούν ως είσοδοι ή έξοδοι όπου αναγνωρίζουν δύο καταστάσεις, είτε 0 είτε 5V, αντιστοιχώντας σε LOW ή HIGH, (ii) Οι αναλογικοί (Analog) ακροδέκτες, που χρησιμοποιούνται ως είσοδοι αποκλειστικά για τη μέτρηση αναλογικών τάσεων από 0 έως 5V, μετατρέποντας τις τιμές αυτές σε αριθμούς από 0 έως 1023, και (iii) Οι ψευδοαναλογικοί PWM μπορούν να χρησιμοποιηθούν ως έξοδοι για τη δημιουργία τετραγωνικών κυματομορφών μεταβλητού κύκλου εργασίας (duty cycle).

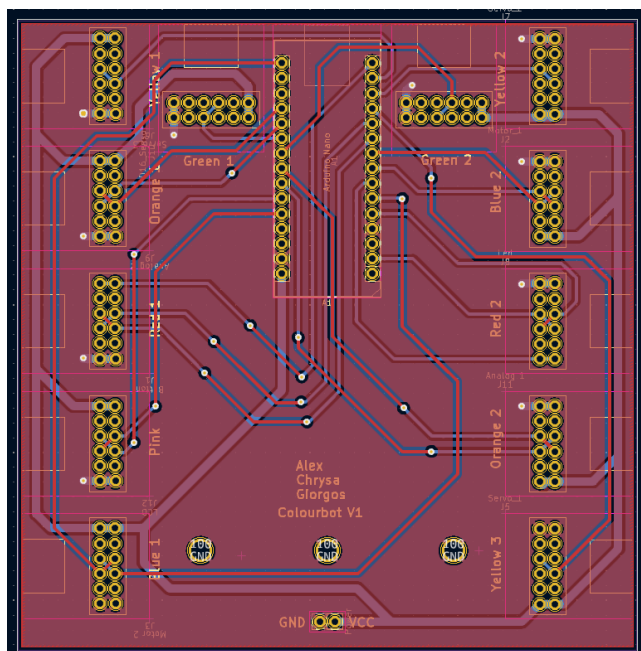
Η κεντρική πλακέτα διαχειρίζεται τη σύνδεση όλων των ηλεκτρονικών εξαρτημάτων με τον Arduino, χρησιμοποιώντας ειδικούς συνδετήρες όπως παρουσιάστηκαν παραπάνω. Η σχεδίαση όλων των πλακετών πραγματοποιήθηκε με το λογισμικό KiCad EDA - Schematic Capture & PCB Design Software [21]. Το KiCad είναι μια ανοιχτού κώδικα σουίτα λογισμικού για τον σχεδιασμό ηλεκτρονικών κυκλωμάτων και την ανάπτυξη πλακετών τυπωμένου κυκλώματος (PCB – Printed Circuit Boards). Χρησιμοποιείται από μηχανικούς, σχεδιαστές και χομπίστες για τη δημιουργία σύνθετων ηλεκτρονικών σχεδίων και την υλοποίηση των κυκλωμάτων τους σε πλακέτες.

Η υλοποίηση ξεκινάει με την σχεδίαση διαγραμμάτων κυκλωμάτων με τη χρήση γραφικών συμβόλων. (schematic diagram, Εικόνα 3.9). Αυτά τα διαγράμματα μπορούν στη συνέχεια να χρησιμοποιηθούν για τη δημιουργία της πλακέτας.



Εικόνα 3-9. Σχηματικό διάγραμμα Κεντρικής Μονάδας.

Αφού λοιπόν δημιουργηθεί το ηλεκτρονικό κύκλωμα, το KiCad προσφέρει εργαλεία για να σχεδιαστεί το φυσικό layout της πλακέτας PCB (Εικόνα 3.10), όπου τοποθετούνται τα εξαρτήματα και καθορίζονται οι διαδρομές των συνδέσεων. Όπως φαίνεται και από την εικόνα, η πλακέτα είναι δύο όψεων. Με κόκκινο χρώμα αποτυπώνονται οι διάδρομοι χαλκού στην πάνω πλευρά της πλακέτας, ενώ με μπλε χρώμα οι διάδρομοι χαλκού στην κάτω πλευρά.



Εικόνα 3-10. Σχέδιο τυπωμένου κυκλώματος Κεντρικής Μονάδας.

Η τελική μορφή μετά και την συγκόλληση των εξαρτημάτων παρουσιάζεται στην Εικόνα 3.11.



Εικόνα 3-11. Κεντρική Μονάδα συστήματος.

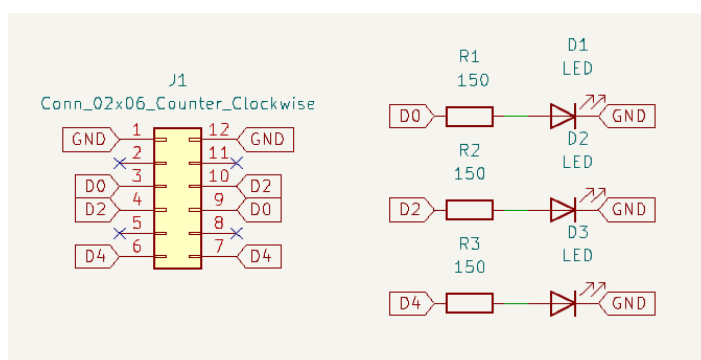
### ***3.4 Αισθητήρες και Επενεργητές***

Όσον αφορά τους αισθητήρες και τους επενεργητές, προκειμένου το σύστημα να είναι λειτουργικό και οι μαθητές να έχουν τη δυνατότητα να δημιουργούν διαφορετικές συσκευές, όχι μόνο περιορισμένες σε ρομπότ αλλά ακόμη και για παράδειγμα ένα έξυπνο σπίτι ή ό,τι άλλο μπορεί να φανταστούν, έχουν αναπτυχθεί ορισμένα ηλεκτρονικά εξαρτήματα. Έχουν σχεδιαστεί προσαρμοσμένα PCBs, έτσι ώστε όλες οι συσκευές να χρησιμοποιούν τον ίδιο τύπο σύνδεσης με τον σύνδεσμο τύπου-C για τροφοδοσία και επικοινωνία με την κεντρική μονάδα. Κάθε σχέδιο περιλαμβάνει επίσης όλα τα απαραίτητα ηλεκτρονικά εξαρτήματα (δηλαδή αντιστάσεις pull down, διαιρέτες τάσης, πυκνωτές κλπ.). Επίσης όλα τα ηλεκτρονικά εξαρτήματα επιλέχθηκαν αυστηρά ώστε να είναι της λογικής των 5V, όπως και η κεντρική μονάδα από την οποία τροφοδοτούνται.

Αρχικά παρουσιάζονται τα εξαρτήματα του συστήματος που είναι Μονάδες Εξόδου (Output Modules). Είναι τμήματα του συστήματος που χρησιμοποιούνται για να διαχειρίζονται και να ελέγχουν τα σήματα ή τα δεδομένα που αποστέλλονται από την κεντρική μονάδα. Στη συνέχεια, παρουσιάζονται τα εξαρτήματα του συστήματος που είναι Μονάδες Εισόδου (Input Modules). Είναι τμήματα του συστήματος που λαμβάνουν σήματα ή δεδομένα από εξωτερικές συσκευές ή αισθητήρες και τα μετατρέπουν σε μορφή κατάλληλη για επεξεργασία από την κεντρική μονάδα.

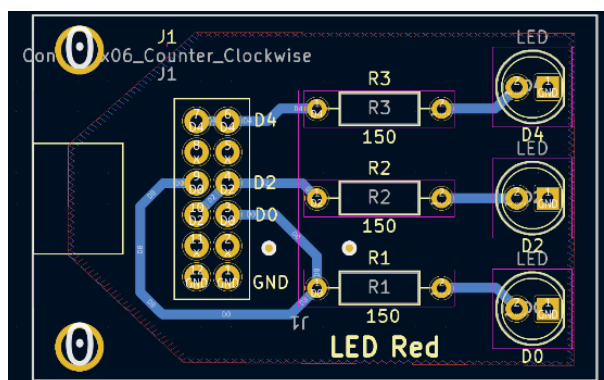
### 3.4.1 Άρθρωμα LED

Το άρθρωμα των LED δίνει στον χρήστη την δυνατότητα ελέγχου τριών LED. Αποτελείται από ένα κόκκινο, ένα κίτρινο και ένα πράσινο LED, τον συνδετήρα τύπου-C και τρεις αντιστάσεις τιμής 150Ω ή μεγαλύτερης, ώστε να διατηρείται το ρεύμα στο κάθε ένα σε ασφαλή επίπεδα. Ανήκει στην κατηγορία του κόκκινου χρωματικού κώδικα και δέχεται 3 ψηφιακά HIGH/LOW σήματα ελέγχοντας πότε το κάθε LED θα εκπέμπει φως. Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.12.



Εικόνα 3-12. Σχηματικό διάγραμμα LED.

Το σχέδιο του τυπωμένου κυκλώματος έγινε μονής όψης καθώς έχει μικρή πολυπλοκότητα (Εικόνα 3.13).



Εικόνα 3-13. Σχέδιο τυπωμένου κυκλώματος LED.

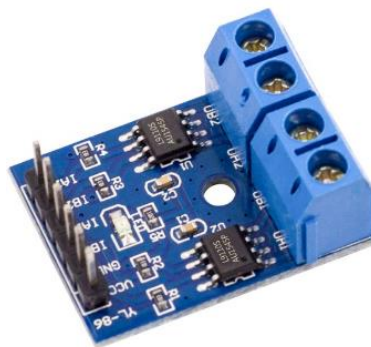
Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.14.



Εικόνα 3-14. Άρθρωμα LED.

### 3.4.2 Άρθρωμα DC Motor

Το άρθρωμα των DC Motor δίνει στον χρήστη την δυνατότητα ελέγχου ενός κινητήρα συνεχούς ρεύματος. Απαρτίζεται από έναν κινητήρα συνεχούς ρεύματος, τον συνδετήρα τύπου C και έναν οδηγό κινητήρα (motor driver) για τον έλεγχο του. Ο motor driver L9110S (Εικόνα 3.15) είναι δύο καναλιών, διαθέτει δυο ανεξάρτητα τσιπ οδήγησης όπου το κάθε ένα μπορεί να οδηγήσει 800mA συνεχούς ρεύματος και να λειτουργήσουν στα 2.5V έως 12V.



Εικόνα 3-15. Motor driver L9110S.

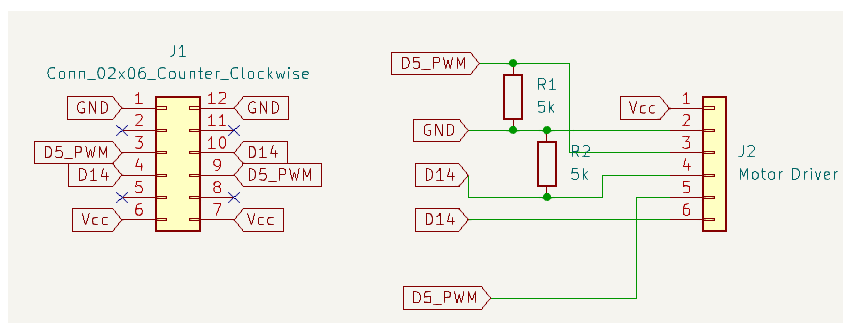
Ο L9110S βασίζεται στην αρχιτεκτονική H-bridge (γέφυρα H) για την οδήγηση κινητήρων. Η αρχιτεκτονική H-bridge επιτρέπει την ελεγχόμενη αλλαγή της πολικότητας της τάσης στους κινητήρες, επιτρέποντας τον έλεγχο της κατεύθυνσης περιστροφής του κινητήρα. Η H-bridge αποτελείται από τέσσερις διακόπτες (συνήθως τρανζίστορ ή MOSFETs) που συνδέονται έτσι ώστε να ελέγχουν τη ροή του ρεύματος μέσα από τον κινητήρα. Ανάλογα με

τη διάταξη των διακοπών, το ρεύμα μπορεί να ρέει προς τη μία κατεύθυνση ή την αντίθετη στον κινητήρα, επιτρέποντας τον έλεγχο της κατεύθυνσης της περιστροφής. Ο κινητήρας (Εικόνα 3.16) συνδέεται με τον οδηγό μέσω δύο σετ βιδωτών ακροδεκτών.



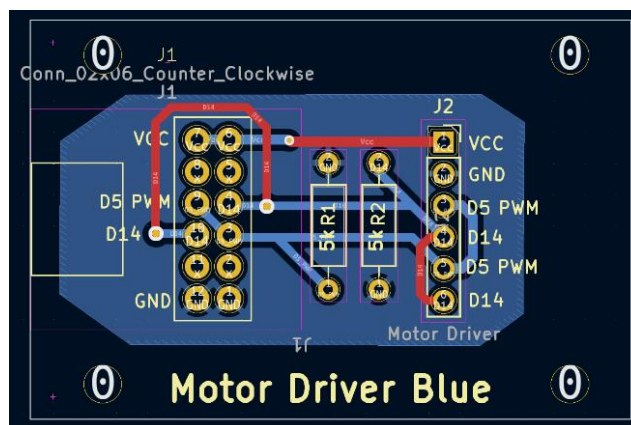
Εικόνα 3-16. Gear Motor TT-130rpm.

Ανήκει στην κατηγορία του μπλε χρωματικού κώδικα, δέχεται 1 ψηφιακό σήμα που χρησιμοποιείται για να αλλάξει την κατεύθυνσή του και 1 σήμα PWM για τον έλεγχο της ταχύτητας του. Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.17.



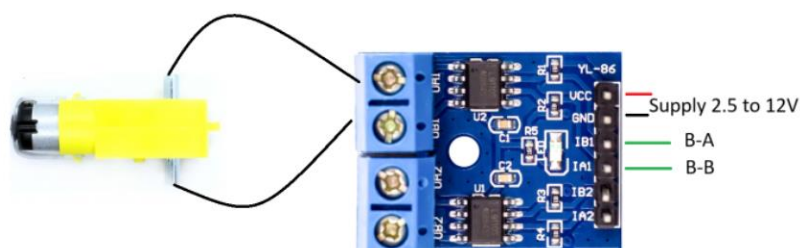
Εικόνα 3-17. Σχηματικό διάγραμμα DC Motor.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.18.



Εικόνα 3-18. Σχέδιο τυπωμένου κυκλώματος DC Motor.

Στον παρακάτω πίνακα παρουσιάζεται η λειτουργία αυτής της πλακέτας, για χρήση ενός κινητήρα όπως και στο παρόν σύστημα (Πίνακας 3.2).



Πίνακας 3-2. Λειτουργία DC Motor Driver.		
B-A	B-B	Motor
LOW	LOW	STOP
HIGH	LOW	FORWARD
LOW	HIGH	REVERSE
HIGH	HIGH	STOP

Έχουν προστεθεί και δυο αντιστάσεις pull-down καθώς διατηρούν τις ψηφιακές εισόδους σε λογικό 0 όταν δεν υπάρχει σήμα, εξασφαλίζοντας ότι το κύκλωμα δεν θα λαμβάνει τυχαίες



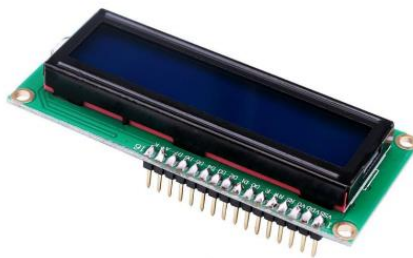
τιμές από θόρυβο ή floating σήματα. Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.19.



Εικόνα 3-19. Άρθρωμα DC Motor.

### 3.4.3 Άρθρωμα LCD

Το άρθρωμα της LCD (οθόνη υγρών κρυστάλλων) δίνει την δυνατότητα στον χρήστη να εμφανίσει στην οθόνη 16 χαρακτήρες σε 2 γραμμές (δηλαδή συνολικά 32 χαρακτήρες). Απαρτίζεται από μια LCD, τον συνδετήρα τύπου C και μια μονάδα σειριακής διασύνδεσης I2C. Κάθε χαρακτήρας στην οθόνη (Εικόνα 3.20) εμφανίζεται σε πίνακα 5x8 pixels. Λειτουργεί στα 5V, υποστηρίζει την τυπική διεπαφή parallel interface μέσω 16 ακίδων και διαθέτει μπλε υπόβαθρο με λευκούς χαρακτήρες, το οποίο καθιστά την οθόνη πολύ ευανάγνωστη ακόμα και σε σκοτεινά περιβάλλοντα.

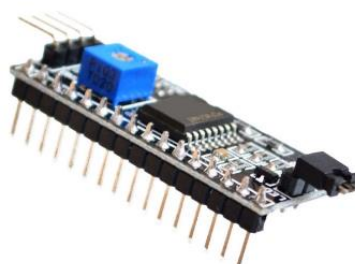


Εικόνα 3-20. Basic 16x2 Character LCD.

Με τη μονάδα διασύνδεσης I2C (Εικόνα 3.21), μπορεί να πραγματοποιηθεί η εμφάνιση δεδομένων στην οθόνη μέσω μόνο 2 καλωδίων. Το LCD Display I2C Interface Module είναι ένας προσαρμογέας που επιτρέπει τη σύνδεση μιας τυπικής LCD οθόνης χρησιμοποιώντας μόνο δύο γραμμές επικοινωνίας μέσω του I2C (Inter-Integrated Circuit) πρωτοκόλλου. Αυτό

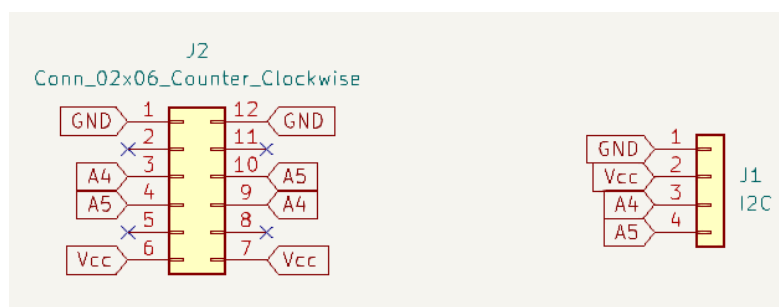


μειώνει τον αριθμό των ακίδων που απαιτούνται για τη σύνδεση της οθόνης με έναν μικροελεγκτή, όπως το Arduino, σε σχέση με την παραδοσιακή παράλληλη διεπαφή (parallel interface).



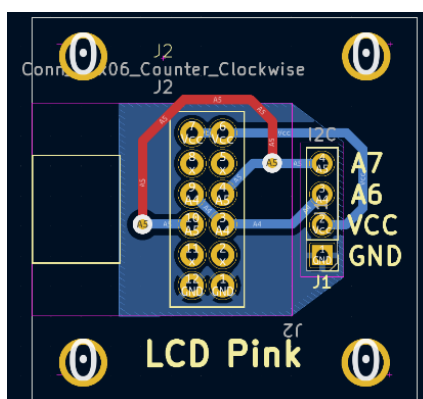
Εικόνα 3-21. LCD Display I2C Interface Module.

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.22.



Εικόνα 3-22. Σχηματικό διάγραμμα LCD.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.23.



Εικόνα 3-23. Σχέδιο τυπωμένου κυκλώματος LCD.

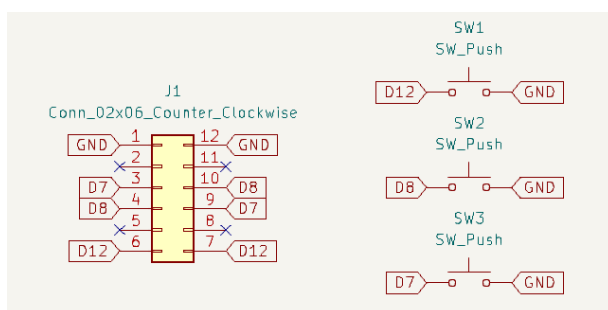
Στην Εικόνα 3.24 αποτυπώνεται ολοκληρωμένο το άρθρωμα με τα επιμέρους στελέχη που το αποτελούν.



Εικόνα 3-24. Άρθρωμα LCD.

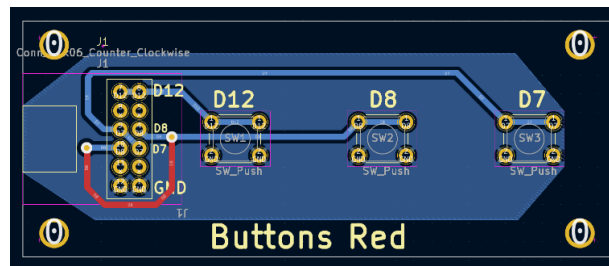
### 3.4.4 Άρθρωμα Buttons

Το άρθρωμα των Buttons αποτελείται από τρεις διακόπτες και τον συνδετήρα τύπου-C. Ο χρήστης έχει στη διάθεσή του μέχρι τρεις ανεξάρτητους διακόπτες για την υλοποίηση εφαρμογών. Ο διακόπτης αυτός ονομάζεται Στιγμιαίος Διακόπτης (Momentary Switch) και παραμένει ενεργοποιημένος μόνο όσο το κουμπί είναι πατημένο. Μόλις το κουμπί απελευθερωθεί, επιστρέφει στην αρχική του θέση. Ανήκει στην κατηγορία Normally Open (NO), το κύκλωμα είναι ανοιχτό όταν δεν πατιέται το κουμπί. Μόνο όταν πατηθεί, κλείνει το κύκλωμα. Στην περίπτωση του προτεινόμενου συστήματος όταν πατηθεί το κουμπί συνδέεται στη γείωση, άρα στέλνει στην κεντρική πλακέτα σήμα LOW και όσο δεν το πατάει ο χρήστης HIGH. Ανήκει στην κατηγορία του κόκκινου χρωματικού κώδικα και αποστέλλει 3 ψηφιακά σήματα. Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.25.



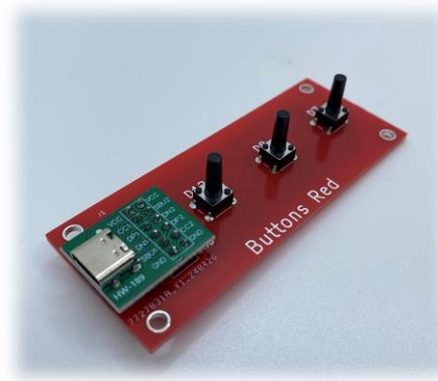
Εικόνα 3-25. Σχηματικό διάγραμμα Buttons.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.26.



Εικόνα 3-26. Σχέδιο τυπωμένου κυκλώματος Buttons.

Στην Εικόνα 3.27 αποτυπώνεται το άρθρωμα ολοκληρωμένο.



Εικόνα 3-27. Άρθρωμα Buttons.

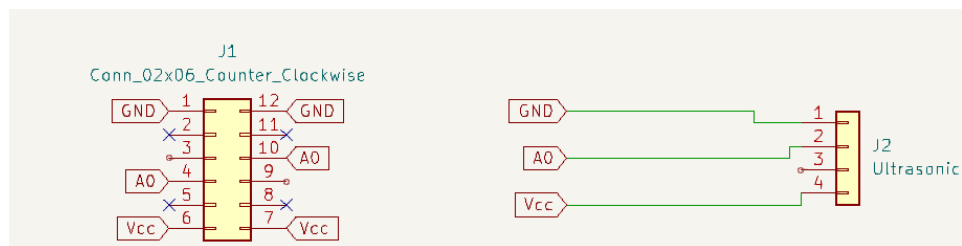
### 3.4.5 Άρθρωμα Sonar Sensor

Το άρθρωμα του Sonar Sensor (αισθητήρας υπερήχων) αποτελείται από τον αναλογικό αισθητήρα URM09 (Εικόνα 3.28) για τον υπολογισμό απόστασης και τον συνδετήρα τύπου-C. Η απόσταση που μπορεί να υπολογίσει είναι από 2 εκ. έως 500 εκ., με ακρίβεια ενός εκατοστού. Η αρχή λειτουργίας των αισθητήρων υπερήχων βασίζεται στη χρήση ηχητικών κυμάτων υψηλής συχνότητας (υπερήχων) για την ανίχνευση αποστάσεων και αντικειμένων. Οι αισθητήρες αυτοί χρησιμοποιούν την εκπομπή και την αντανάκλαση των υπερηχητικών κυμάτων για να υπολογίσουν την απόσταση ενός αντικειμένου, με τρόπο παρόμοιο με τη λειτουργία ενός σόναρ ή ενός ραντάρ υπερήχων. Τα βασικά στάδια λειτουργίας τους είναι η εκπομπή υπερηχητικών κυμάτων, η αντανάκλαση των κυμάτων, ο χρονισμός της επιστροφής και ο υπολογισμός απόστασης. Ανήκει στην κατηγορία του πορτοκαλί χρωματικού κώδικα και αποστέλλει 1 αναλογικό σήμα.



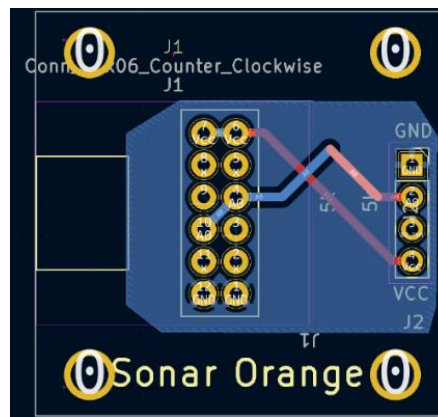
Εικόνα 3-28. Αισθητήρας υπερήχων URM09.

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.29.



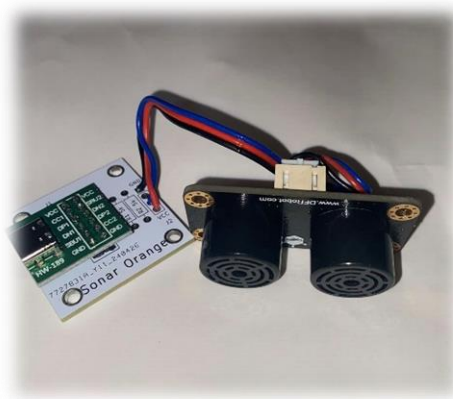
Εικόνα 3-29. Σχηματικό διάγραμμα Sonar.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.30.



Εικόνα 3-30. Σχέδιο τυπωμένου κυκλώματος Sonar.

Η τελική μορφή του αρθρώματος παρουσιάζεται στην Εικόνα 3.31.



Εικόνα 3-31. Άρθρωμα Sonar.

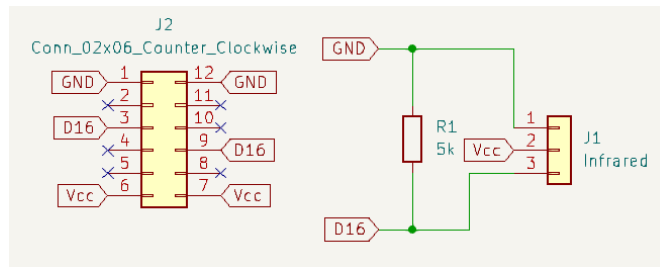
### 3.4.6 Άρθρωμα Infrared Sensor

Το άρθρωμα του Infrared Sensor (αισθητήρας υπέρυθρων) αποτελείται από τον αισθητήρα υπέρυθρων (Εικόνα 3.32) για την ανίχνευση ενός αντικειμένου, μια αντίσταση pull-down και τον συνδετήρα τύπου-C. Το αποτέλεσμα του είναι HIGH/LOW ανάλογα με το εάν ανίχνευσε κάποιο αντικείμενο. Η απόσταση που μπορεί να ανιχνεύσει είναι από  $10 \pm 5$  mm έως  $80 \pm 20$  mm. Η αρχή λειτουργίας του βασίζεται στην ανίχνευση της υπέρυθρης ακτινοβολίας που εκπέμπεται από αντικείμενα. Όλα τα αντικείμενα που έχουν θερμοκρασία πάνω από το απόλυτο μηδέν (0 K) εκπέμπουν υπέρυθρη ακτινοβολία, η οποία μπορεί να ανιχνευτεί από αυτούς τους αισθητήρες. Οι αισθητήρες υπέρυθρων χρησιμοποιούνται σε διάφορες εφαρμογές, όπως η ανίχνευση κίνησης, η μέτρηση θερμοκρασίας, και η παρακολούθηση αντικειμένων. Ανήκει στην κατηγορία του πράσινου χρωματικού κώδικα και αποστέλλει 1 ψηφιακό σήμα.



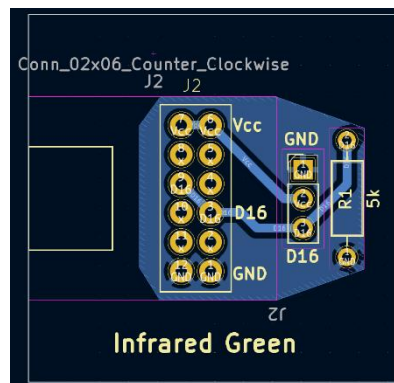
Εικόνα 3-32. Αισθητήρας υπέρυθρων.

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.33.



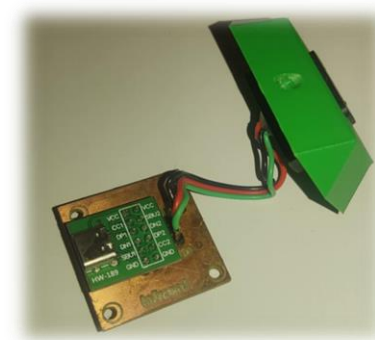
Εικόνα 3-33. Σχηματικό διάγραμμα Infrared.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος παρουσιάζεται στην Εικόνα 3.34.



Εικόνα 3-34. Σχέδιο τυπωμένου κυκλώματος Infrared.

Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.35.



Εικόνα 3-35. Άρθρωμα Infrared.

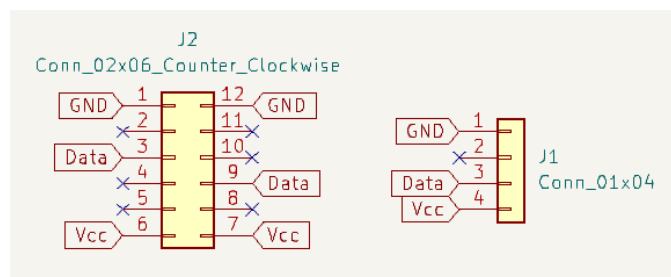
### 3.4.7 *Αρθρώμα Temperature & Humidity Sensor*

Το άρθρωμα του Temperature & Humidity Sensor (αισθητήρας θερμοκρασίας και υγρασίας) αποτελείται από τον αισθητήρα θερμοκρασίας και υγρασίας AM2302 (Εικόνα 3.36) και τον συνδετήρα τύπου-C. Ο AM2302 είναι ένας αισθητήρας υψηλής ακρίβειας και είναι αρκετά δημοφιλής για εφαρμογές σε πλατφόρμες Arduino και Raspberry Pi λόγω της ευκολίας χρήσης του. Το εύρος μέτρησης θερμοκρασίας είναι από  $-40^{\circ}\text{C}$  έως  $+80^{\circ}\text{C}$  με ακρίβεια  $\pm 0.5^{\circ}\text{C}$  και ανάλυση  $0.1^{\circ}\text{C}$ . Το εύρος μέτρησης υγρασίας είναι 0-100% RH (Relative Humidity - Σχετική Υγρασία) με ακρίβεια  $\pm 2\%$  RH (από 0-80% RH) και ανάλυση 0.1% RH. Χρησιμοποιεί ψηφιακή επικοινωνία (μονοκατευθυντική) για την αποστολή δεδομένων. Ο AM2302 διαθέτει ένα εσωτερικό θερμίστορ για τη μέτρηση της θερμοκρασίας και έναν αισθητήρα χωρητικότητας για τη μέτρηση της υγρασίας. Αυτά τα δεδομένα μετατρέπονται σε ψηφιακή μορφή και στέλνονται ως ένας 40-bit σειριακός αριθμός στο μικροελεγκτή. Ανήκει στην κατηγορία του κίτρινου χρωματικού κώδικα και αποστέλλει 1 ψηφιακό σήμα.



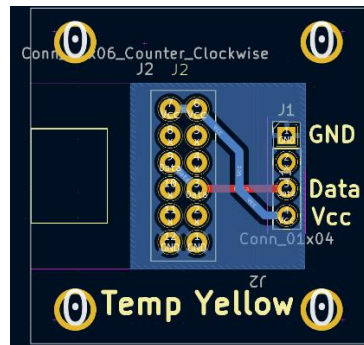
Εικόνα 3-36. Αισθητήρας θερμοκρασίας και υγρασίας AM2302.

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.37.



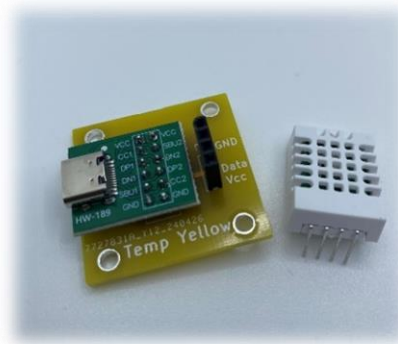
Εικόνα 3-37. Σχηματικό διάγραμμα Temperature & humidity.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.38.



Εικόνα 3-38. Σχέδιο τυπωμένου κυκλώματος Temperature & humidity.

Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.39.



Εικόνα 3-39. Άρθρωμα Temperature & humidity.

### 3.4.8 Άρθρωμα Line Tracking

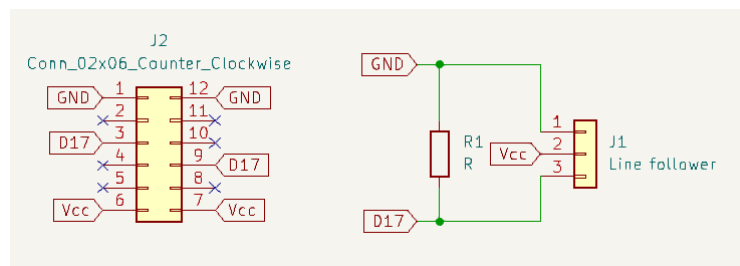
Το άρθρωμα του line tracking αποτελείται από έναν αισθητήρα υπέρυθρων για την ανίχνευση διαδρομής, μια αντίσταση pull-down και τον συνδετήρα τύπου-C. Ο αισθητήρας υπέρυθρων TCRT5000 (Εικόνα 3.40) χρησιμοποιείται για την ανίχνευση χρώματος. Εκπέμπει υπέρυθρες και ανιχνεύει αν λαμβάνει την αντανάκλαση. Χρησιμοποιείται συχνά σε ρομπότ που ακολουθούν μια γραμμή διαδρομής, όπου αυτόματα γίνεται καταγραφή των δεδομένων που δέχεται ο αισθητήρας και έτσι μπορεί να ανιχνεύσει εάν μια επιφάνεια είναι λευκή ή μαύρη. Η απόσταση μέτρησης κυμαίνεται από 1mm έως 8mm και το κεντρικό σημείο είναι περίπου 2.5mm. Το αποτέλεσμα του είναι HIGH/LOW ανάλογα αν ανίχνευσε γραμμή ή όχι. Ανήκει στην κατηγορία του πράσινου χρωματικού κώδικα και αποστέλλει 1 ψηφιακό σήμα.





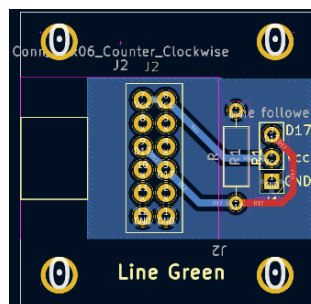
Εικόνα 3-40. Αισθητήρας υπέρυθρων TCRT50.

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.41.



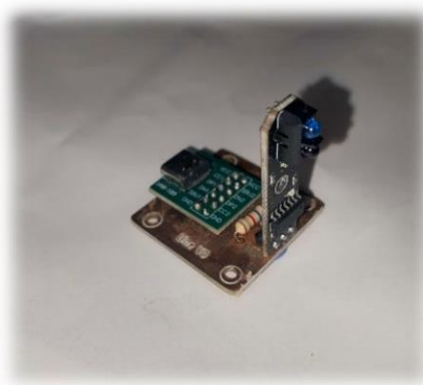
Εικόνα 3-41. Σχηματικό διάγραμμα Line Tracking.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.42.



Εικόνα 3-42. Σχέδιο τυπωμένου κυκλώματος Line Tracking.

Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.43.



Εικόνα 3-43. Άρθρωμα Line Tracking.

### 3.4.9 Άρθρωμα Joystick

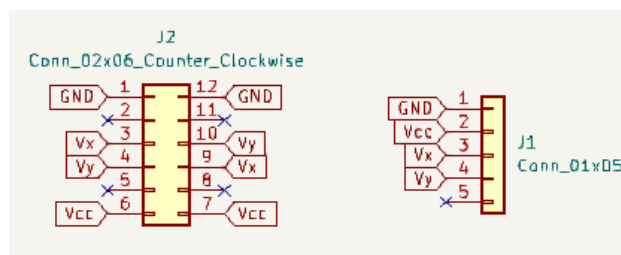
Το άρθρωμα του Joystick αποτελείται από ένα Thumb Joystick Module και τον συνδετήρα τύπου-C. Το Thumb Joystick Module (Εικόνα 3.44) είναι ένας μικρός αναλογικός ελεγκτής τύπου joystick, που μοιάζει πολύ με τα χειριστήρια των παιχνιδοκονσολών. Αυτό το είδος joystick χρησιμοποιείται σε διάφορες ηλεκτρονικές εφαρμογές για την παροχή διπλού αναλογικού ελέγχου (σε δύο άξονες), καθώς και ενός κουμπιού όταν πιέζεται προς τα κάτω (το κουμπί στο παρόν σύστημα δεν χρησιμοποιείται). Το joystick προσφέρει αναλογική έξοδο για τους δύο άξονες: τον οριζόντιο (X) και τον κατακόρυφο (Y). Καθώς κινείται ο μοχλός σε διαφορετικές κατευθύνσεις, οι τιμές τάσης των σημάτων στον άξονα X και Y αλλάζουν αναλογικά. Όταν το joystick βρίσκεται στην κεντρική θέση, οι τάσεις στις εξόδους X και Y είναι περίπου στο μισό της τάσης τροφοδοσίας (συνήθως 2.5V εάν τροφοδοτείται στα 5V). Η κεντρική θέση σημαίνει ότι δεν υπάρχει κίνηση προς κάποια κατεύθυνση. Ανήκει στην κατηγορία του πορτοκαλί χρωματικού κώδικα και αποστέλλει 2 αναλογικά σήματα.



Εικόνα 3-44. Thumb Joystick Module.

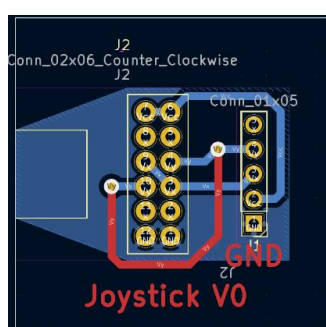
---

Το σχηματικό διάγραμμα του αρθρώματος παρουσιάζεται στην Εικόνα 3.45.



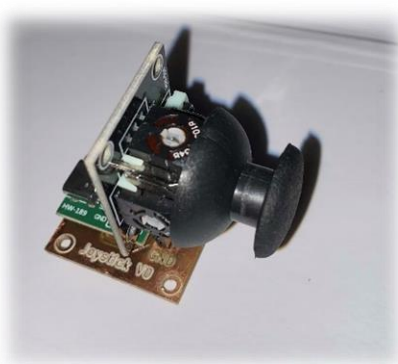
Εικόνα 3-45. Σχηματικό διάγραμμα Joystick.

Το σχέδιο του τυπωμένου κυκλώματος του αρθρώματος είναι διπλής όψης και παρουσιάζεται στην Εικόνα 3.46.



Εικόνα 3-46. Σχέδιο τυπωμένου κυκλώματος Joystick.

Η τελική μορφή του αρθρώματος φαίνεται στην Εικόνα 3.47.



Εικόνα 3-47. Άρθρωμα Joystick.

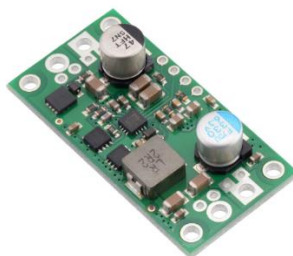
### 3.5 Μονάδα Ισχύος

Για να τροφοδοτηθούν εξωτερικά οι συσκευές που σχεδιάστηκαν, αναπτύχθηκε μια μονάδα ισχύος, που παρέχει ενέργεια στο σύστημα, συμπεριλαμβανομένων της κύριας πλακέτας και όλων των άλλων συνδεδεμένων συσκευών. Έχουν χρησιμοποιηθεί αυτοτελή

εξαρτήματα και έχει συναρμολογηθεί μια μονάδα μπαταρίας με 4 κελιά (μπαταρίες τύπου 18650, Εικόνα 3.48) και ένας μετατροπέας τάσης (Εικόνα 3.49), για να μετατρέπεται η τάση εισόδου των μπαταριών σε 5V για το Arduino και τους υπόλοιπους αισθητήρες.



Εικόνα 3-48. Μπαταριοθήκη 4x18650 - με Καλώδια.



Εικόνα 3-49. Pololu 5V, 6A Step-Down Voltage Regulator D24V60F5.

Η επιλογή αυτών των μπαταριών (Εικόνα 3.50) γίνεται λόγω της υψηλής ενεργειακής τους πυκνότητας και της δυνατότητας τους για πολλαπλές επαναφορτίσεις. Επιπλέον, προσφέρουν μια ελαφριά στοίβα μπαταρίας καθιστώντας ολόκληρο το κέλυφος αρκετά ελαφρύ ώστε να υποστηρίζει την συνεχή τροφοδοσία φορητών συσκευών για περίπου 3 έως 4 ώρες (ανάλογα με τις συνδεδεμένες συσκευές).

Οι μπαταρίες αυτές έχουν τεχνολογία INR. Το INR αναφέρεται σε Lithium Nickel Manganese Cobalt Oxide ( $\text{LiNiMnCoO}_2$ ). Αυτή η χημεία ιόντων λιθίου παρέχει ένα ισορροπημένο μείγμα απόδοσης με αυξημένη ασφάλεια, μεγάλη ενεργειακή πυκνότητα, σταθερότητα υπό φορτίο και καλές επιδόσεις σε υψηλά ρεύματα εκφόρτισης. Κάθε μπαταρία έχει χωρητικότητα 3200mAh (milliampere-hour), που σημαίνει ότι μπορεί να παρέχει 3200mA για μία ώρα πριν αδειάσει πλήρως ή 1600mA για δύο ώρες, κ.λπ. Οι μπαταρίες 18650 συνήθως έχουν ονομαστική τάση 3.6V ή 3.7V όταν είναι πλήρως φορτισμένες και μπορούν να φτάσουν έως περίπου 4.2V πλήρως φορτισμένες και να εκφορτιστούν έως περίπου 2.5-3.0V. Έχουν μεγάλη διάρκεια ζωής, περίπου 300-500 κύκλους φόρτισης/εκφόρτισης, ανάλογα με τις συνθήκες χρήσης.



Εικόνα 3-50. Μπαταρία Λιθίου 18650 3.6V 3200mAh – INR.

Είναι συνδεδεμένες σε σειρά. Οι μπαταρίες συνδέονται από άκρη με άκρη (θετικό με αρνητικό) ώστε η τάση να προστίθεται, αλλά η χωρητικότητα να παραμένει η ίδια. Η τάση είναι  $4 \times 3.6V = 14.4V$  ενώ η χωρητικότητα παραμένει στα 3200mAh.

Ο μετατροπέας τάσης step-down (ή Buck converter) δέχεται είσοδο τάσης έως 38 V και την μειώνει αποτελεσματικά στα 5 V, με διαθέσιμο ρεύμα εξόδου περίπου 6 A. Τυπικές αποδόσεις από 80% έως 95% καθιστούν αυτόν τον μετατροπέα κατάλληλο για εφαρμογές υψηλής ισχύος, όπως η τροφοδοσία κινητήρων ή σερβομηχανισμών, ενώ οι υψηλές αποδόσεις διατηρούνται ακόμη και σε χαμηλά φορτία με τη δυναμική αλλαγή της συχνότητας μεταγωγής.

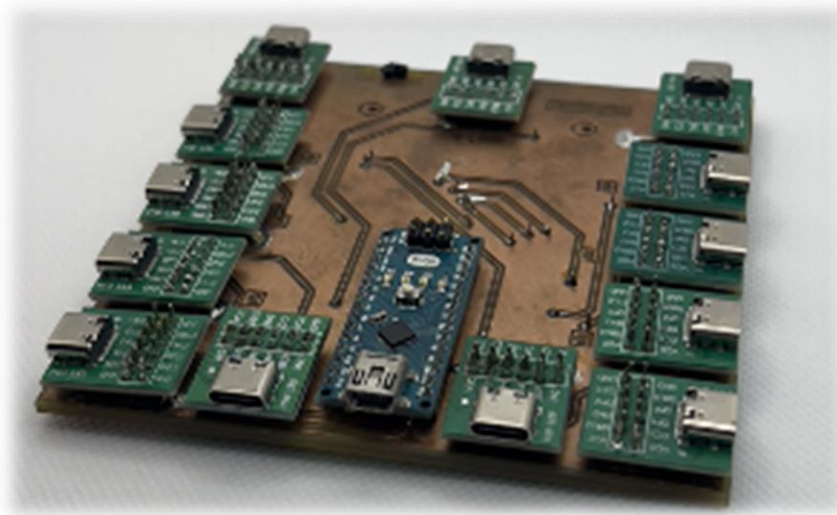
Στην Εικόνα 3.51 αποτυπώνεται ολοκληρωμένο το άρθρωμα της μονάδας ισχύος με τα επιμέρους στελέχη που το αποτελούν.



Εικόνα 3-51. Μονάδα Ισχύος.

### 3.6 Πρωτότυπες πλακέτες

Οι πλακέτες που δημιουργήθηκαν, όπως αναφέρθηκε, στάλθηκαν σε εξωτερική επιχείρηση για την κατασκευή τους. Ωστόσο, είχε πραγματοποιηθεί μια πρώτη κατασκευή τους στο εργαστήριο Ευφών Συστημάτων και Ρομποτικής, της Σχολής Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης<sup>1</sup>. Η κατασκευή των πρώτων πρωτοτύπων μέσω έγινε με χρήση του λογισμικού ‘FlatCam for Gcode production’ [22]. Το FlatCAM είναι ένα ανοιχτού κώδικα λογισμικό που χρησιμοποιείται για την προετοιμασία αρχείων G-code από αρχεία Gerber και Excellon, τα οποία είναι μια κοινή φόρμα δεδομένων για τον σχεδιασμό πλακετών τυπωμένων κυκλωμάτων (PCB). Τα G-code αρχεία είναι απαραίτητα για τον έλεγχο CNC μηχανών. Αυτό γίνεται μέσω της διαμόρφωσης μονοπατιών για τη χάραξη των ιχνών, την κοπή και τη διάτρηση του PCB. Για την κατασκευή χρησιμοποιήθηκε ειδική συσκευή CNC.



Εικόνα 3-52. Πρωτότυπο Κεντρικής Πλακέτας.

---

<sup>1</sup> Η κατασκευή πραγματοποιήθηκε από τον συνεργάτη του εργαστηρίου κ. Σιούτη Αλέξανδρο.

## ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ArduBlockly

---

Το λογισμικό του ColourBot είναι μια αναβαθμισμένη και προσαρμοσμένη έκδοση του ArduBlockly [23]. Το βασικό του πλεονέκτημα είναι η ευκολία στη χρήση και η δυνατότητα δημιουργίας κώδικα Arduino μέσω οπτικών μπλοκ με λειτουργία drag-and-drop, τα οποία μπορούν εύκολα να μεταφερθούν σε μια πλακέτα Arduino. Ακολουθήθηκε μια παρόμοια προσέγγιση και σε προηγούμενες εργασίες [4, 5], αλλά με μια σημαντική διαφοροποίηση. Σχεδιάστηκε η διεπαφή λαμβάνοντας υπόψη ένα κοινό με ποικίλα χαρακτηριστικά και εκπαιδευτικά επίπεδα, δηλαδή μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης. Ανάλογα με το εκπαιδευτικό επίπεδο του χρήστη, παράχθηκαν διαφορετικές εκδόσεις.

Δημιουργήθηκαν προσαρμοσμένες βιβλιοθήκες για να κατασκευαστούν μπλοκ που ταιριάζουν στα υλικά αρθρώματα του συστήματος. Ο χρήστης επιλέγει τα μπλοκ που χρειάζεται για το έργο του, τα σύρει, τα συνδέει και επιλέγει τις κατάλληλες παραμέτρους για το καθένα. Ταυτόχρονα, το πρόγραμμα μεταγλωττίζει τον απαραίτητο κώδικα C++ για Arduino. Ο κώδικας μπορεί να είναι άμεσα προσβάσιμος μέσω του Arduino IDE, δίνοντας έτσι στον μαθητή ή τον εκπαιδευτικό τη δυνατότητα να δει τον κώδικα παράλληλα με τα οπτικά μπλοκ, διευκολύνοντας έτσι τη διαδικασία εκμάθησης του προγραμματισμού.

### 4.1 Google Blockly

Το Google Blockly [17] είναι μια οπτική βιβλιοθήκη προγραμματισμού που επιτρέπει στους χρήστες να δημιουργούν κώδικα με τη χρήση block-based διεπαφής, δηλαδή με το "σύρσιμο και σύνδεση" (drag-and-drop) μπλοκ εντολών. Πρόκειται για ένα εργαλείο που βοηθάει να διδαχθεί ο προγραμματισμός με έναν πιο διαισθητικό τρόπο, ιδιαίτερα χρήσιμο για αρχάριους και εκπαιδευτικά περιβάλλοντα.

Η διεπαφή του επιτρέπει τη σύνθεση του κώδικα μέσω της σύνδεσης οπτικών μπλοκ. Κάθε μπλοκ αντιπροσωπεύει μια εντολή, έναν βρόχο, μια μεταβλητή ή μια συνάρτηση. Αυτά τα μπλοκ έχουν συγκεκριμένες μορφές και χρώματα για να δείξουν τη λειτουργία τους, διευκολύνοντας την κατανόηση της δομής του προγράμματος. Παρόλο που ο κώδικας

δημιουργείται μέσω μπλοκ, το Blockly μπορεί να μετατρέψει αυτόν τον οπτικό κώδικα σε κώδικα σε διάφορες γλώσσες προγραμματισμού όπως JavaScript, Python, Lua, PHP ή Dart. Αυτό σημαίνει ότι οι μαθητές ή οι αρχάριοι μπορούν να μάθουν τη δομή και τη λογική του προγραμματισμού με οπτικό τρόπο και στη συνέχεια να δουν πώς αυτός ο κώδικας "μεταφράζεται" σε πραγματική γλώσσα προγραμματισμού. Το Blockly χρησιμοποιείται ευρέως στην εκπαίδευση, καθώς επιτρέπει σε μαθητές και αρχάριους να εξοικειωθούν με έννοιες όπως βρόχοι, διακλαδώσεις, συνθήκες και μεταβλητές. Δίνει στους μαθητές τη δυνατότητα να κατανοήσουν τη λογική του προγραμματισμού χωρίς να πρέπει να μάθουν τη σύνταξη μιας γλώσσας εξ αρχής.

Οι προγραμματιστές μπορούν να δημιουργήσουν τα δικά τους προσαρμοσμένα μπλοκ για συγκεκριμένες λειτουργίες ή έργα, επιτρέποντας έτσι μεγάλη ευελιξία και προσαρμογή στις ανάγκες του έργου τους. Το Blockly είναι γραμμένο σε JavaScript και μπορεί να ενσωματωθεί σε οποιοδήποτε πρόγραμμα περιήγησης. Παρέχει ένα GUI περιβάλλον που μπορεί να προσαρμοστεί για διάφορες γλώσσες και ανάγκες. Επίσης, προσφέρει API για προγραμματιστές, ώστε να μπορούν να ενσωματώσουν τη βιβλιοθήκη στις δικές τους εφαρμογές και να δημιουργήσουν νέα μπλοκ και λειτουργικότητα.

Το Blockly δημιουργήθηκε με σαφή προσανατολισμό στη διδασκαλία εννοιών πληροφορικής. Όμως η χρήση του δεν περιορίζεται μόνο για την εκπαίδευση, αλλά μπορεί να αξιοποιηθεί και ως εργαλείο δημιουργίας κώδικα. Σε καμία περίπτωση δεν μπορεί να θεωρηθεί γλώσσα προγραμματισμού, ούτε δημιουργήθηκε ως τέτοια. Τρέχει μέσω browser και δεν χρειάζεται κάποιου είδους εγκατάσταση και αυτός είναι ο λόγος που δεν θεωρείται εφαρμογή.

## 4.2 Ardublockly

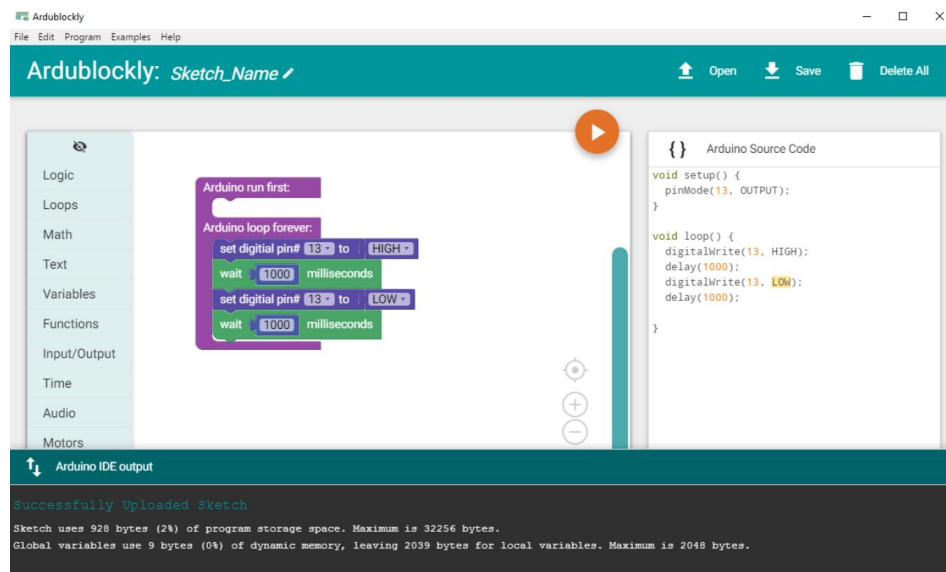
Το Ardublockly<sup>2</sup> είναι ένα εργαλείο βασισμένο στο Google Blockly, σχεδιασμένο για τον οπτικό προγραμματισμό του Arduino. Δημιουργήθηκε από τον προγραμματιστή Carlos Pereira Atencio το 2015 [24]. Όπως και το Blockly, το Ardublockly επιτρέπει στους χρήστες να δημιουργούν κώδικα χρησιμοποιώντας μπλοκ εντολών αντί για γραμμές κώδικα. Είναι συμβατό με τις περισσότερες αυθεντικές πλακέτες Arduino και λειτουργεί σε Windows, Linux και Mac OS X. Οι χρήστες δημιουργούν προγράμματα με drag-and-drop, τα οποία

---

<sup>2</sup> Λογισμικό διαθέσιμο με άδεια Creative Commons Attribution 4.0 International License <https://creativecommons.org/licenses/by/4.0/>



αντιπροσωπεύουν βασικές λειτουργίες του Arduino, όπως είσοδος/έξοδος, αισθητήρες, λειτουργίες LED, μοτέρ, και πολλά άλλα.



Εικόνα 4-1. Περιβάλλον Ardublockly.

Κάθε φορά που ο χρήστης τοποθετεί μπλοκ στην οπτική διεπαφή, το Ardublockly δημιουργεί αυτόματα τον αντίστοιχο κώδικα C++ που χρησιμοποιείται από το Arduino IDE. Ο κώδικας που παράγεται μπορεί να αντιγραφεί και να επικολληθεί ή να αποσταλεί απευθείας στο Arduino μέσω του IDE. Παρέχει ένα εύκολο περιβάλλον για τη μετάβαση από τον οπτικό προγραμματισμό στον πραγματικό κώδικα, καθώς οι χρήστες μπορούν να δουν πώς οι μπλοκ εντολές μετατρέπονται σε κώδικα που εκτελείται στην πλακέτα. Είναι ένα ιδανικό εργαλείο για εκπαιδευτικά έργα STEM (Science, Technology, Engineering, Mathematics), όπου οι μαθητές μπορούν να συνδυάσουν προγραμματισμό και φυσική κατασκευή (π.χ., ρομπότ, έξυπνες συσκευές).

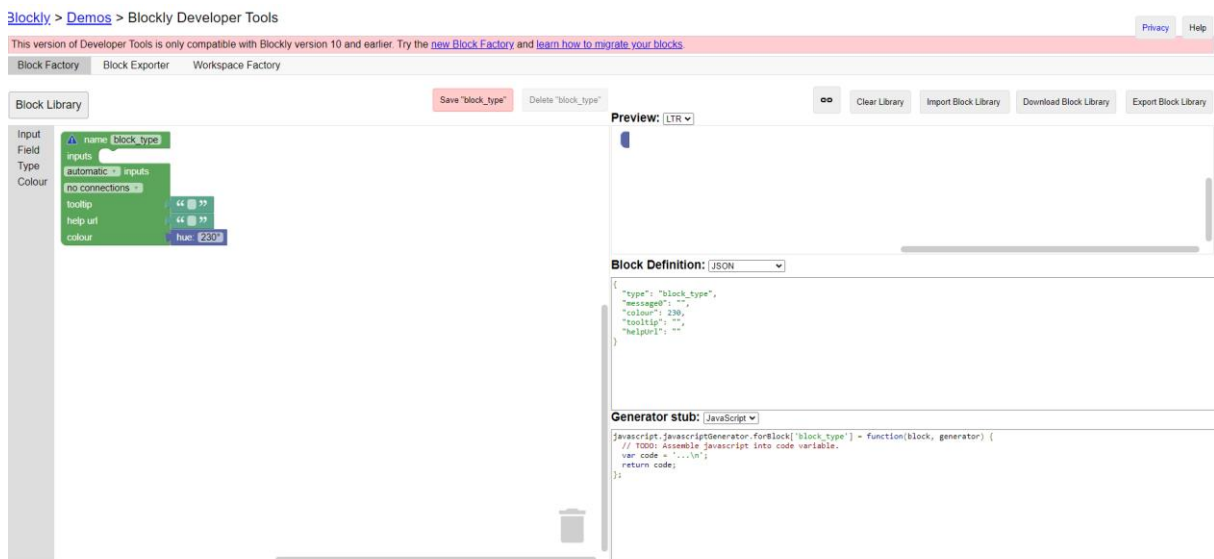
Το λογισμικό του συστήματος αποτελεί μια εξελιγμένη έκδοση του αυθεντικού Ardublockly, στην οποία έχουν τροποποιηθεί τα μπλοκ, με αφαίρεση και προσθήκη στοιχείων, προκειμένου να καλύπτει τις ιδιαίτερες ανάγκες που προκύπτουν από το αναπτυγμένο hardware.

### ***4.3 Εργαλείο Block Factory***

Το Block Factory είναι ένα εργαλείο που παρέχει το Blockly και χρησιμοποιείται για τη δημιουργία προσαρμοσμένων μπλοκ και ανοίγει μέσω browsers. Εάν ο χρήστης θέλει να

δημιουργήσει ένα δικό του μπλοκ εντολών για χρήση σε περιβάλλοντα όπως το Ardublockly, το Blockly, ή άλλες εφαρμογές που υποστηρίζουν block-based προγραμματισμό, το Block Factory παρέχει αυτή τη δυνατότητα. Επιτρέπουν στους προγραμματιστές ή στους εκπαιδευτές να ορίσουν τις λειτουργίες που θέλουν να έχουν τα μπλοκ που δημιουργούν.

Το Block Factory επιτρέπει στον χρήστη να σχεδιάσει οπτικά ένα μπλοκ. Μπορεί να καθορίσει το σχήμα του, τα χρώματα, καθώς και τα σημεία στα οποία θα συνδέεται με άλλα μπλοκ. Μπορεί να ορίσει ποια δεδομένα θα δέχεται το μπλοκ, όπως αριθμούς, κείμενο, λογικές τιμές, κ.λπ. Επίσης, μπορεί να επιλέξει αν το μπλοκ θα περιλαμβάνει άλλες παραμέτρους ή λειτουργίες (π.χ. μεταβλητές, εντολές εισόδου/εξόδου). Μετά τη σχεδίαση του μπλοκ, το εργαλείο δημιουργεί αυτόματα τον κώδικα JavaScript που αντιστοιχεί σε αυτό το μπλοκ. Αυτός ο κώδικας μπορεί να προστεθεί στη βιβλιοθήκη του Blockly, έτσι ώστε να μπορεί να χρησιμοποιηθεί στο δικό του έργο. Τα μπλοκ που δημιουργούνται μπορούν να εξαχθούν ως αρχεία και να χρησιμοποιούνται σε άλλα έργα ή να μοιράζονται με άλλους χρήστες.



Εικόνα 4-2. Περιβάλλον Block Factory.

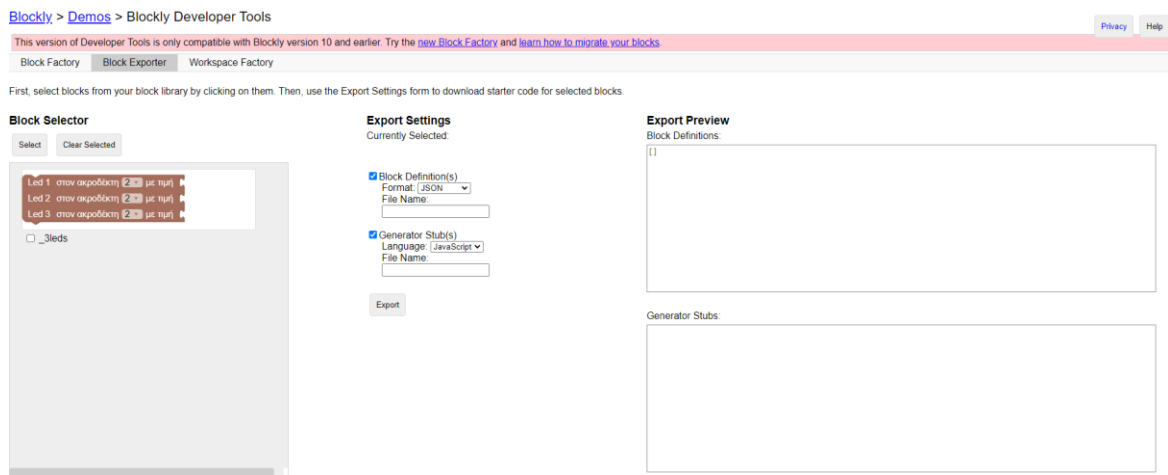
Στο αριστερό μέρος της οθόνης (Εικόνα 4.2) εμφανίζεται ένα μπλοκ πράσινου χρώματος με πολλαπλές υποδοχές, πάνω στο οποίο «χτίζεται» το καινούργιο μπλοκ. Το υπό δημιουργία μπλοκ εμφανίζεται στο δεξί-πάνω μέρος της οθόνης (Πεδίο Preview). Το δεξί μέρος της οθόνης περιλαμβάνει επίσης δύο ακόμα πεδία (Block Definition και Generator Stub), τα οποία είναι υπεύθυνα για τη δημιουργία κώδικα (Code Generator) παράλληλα με τη δημιουργία του μπλοκ. Στο πεδίο Block Definition παράγεται κώδικας που αφορά το φυσικό σχεδιασμό του μπλοκ (χρώμα, σχήμα, τύπος κ.ά.). Ο χρήστης έχει τη δυνατότητα επιλογής γλώσσας

παραγωγής κώδικα (από κάποιες διαθέσιμες). Το πεδίο Generator Stub προορίζεται για την εισαγωγή κώδικα, ο οποίος θα εκτελείται όταν χρησιμοποιείται το παρόν μπλοκ στα πλαίσια κάποιας εφαρμογής. Ο δημιουργός πρέπει να τοποθετεί τα μπλοκ του στην Block Library και στο τέλος να χρησιμοποιεί την επιλογή Download Block Library, η οποία εξάγει αρχεία με κώδικα σε γλώσσα XML (.xml).

## ***4.4 Εργαλείο Block Exporter***

Το Block Exporter στο Blockly είναι ένα εργαλείο που επιτρέπει στους χρήστες να εξάγουν μπλοκ ή ομάδες μπλοκ από ένα έργο σε διάφορες μορφές, ώστε να μπορούν να τα χρησιμοποιήσουν ξανά ή να τα μοιραστούν με άλλους. Η δυνατότητα εξαγωγής μπλοκ, είναι εξαιρετικά χρήσιμη, για όσους θέλουν να αποθηκεύσουν προσαρμοσμένα μπλοκ ή να τα χρησιμοποιήσουν σε άλλες εφαρμογές. Το Block Exporter εξάγει τα μπλοκ σε μορφή XML. Αυτή η μορφή είναι συμβατή με τα περισσότερα εργαλεία που χρησιμοποιούν το Blockly, όπως το Ardublockly, και μπορεί εύκολα να εισαχθεί πίσω σε άλλο έργο ή ακόμα και να επεξεργαστεί χειροκίνητα. Παρέχει επίσης τη δυνατότητα να εισαχθούν XML αρχεία μπλοκ, που έχουν δημιουργηθεί ή μοιραστεί, πίσω στο περιβάλλον του Blockly. Έτσι, μπορεί να συνεχιστεί ένα έργο ή να χρησιμοποιηθούν επαναχρησιμοποιούμενα τμήματα κώδικα. Ο χρήστης μπορεί να δημιουργήσει μια βιβλιοθήκη από μπλοκ και να την εξάγει, ώστε να τη χρησιμοποιεί ξανά και ξανά σε διαφορετικά έργα, βελτιώνοντας τη συνέχεια και την αποδοτικότητα στον προγραμματισμό.

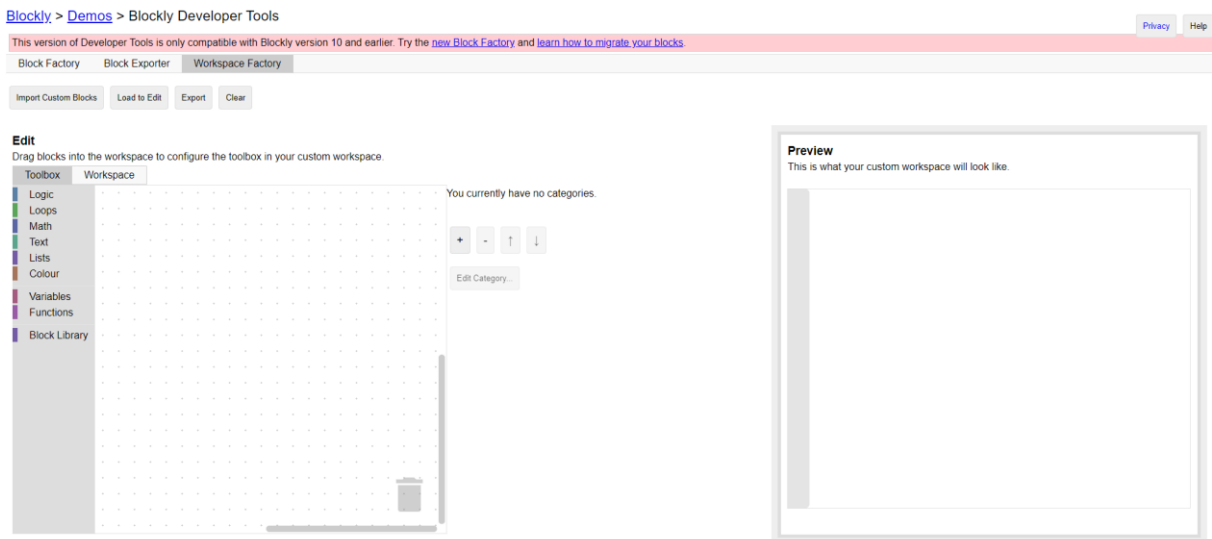
Ο χρήστης πρέπει απλά να επιλέξει ποια από τα μπλοκ, που περιέχονται στην βιβλιοθήκη του, θέλει να χρησιμοποιήσει. Υπάρχει η δυνατότητα εξαγωγής κώδικα υπό μορφή πεδίων (Block Definitions και Generator Stubs) ή υπό μορφή αρχείων σε διάφορες κειμενικές γλώσσες (JavaScript, Python, PHP, Lua, Dart). Το εργαλείο δημιουργεί δύο τύπους αρχείων, ένα αρχείο που περιλαμβάνει κώδικα που σχετίζεται με τη φυσική σχεδίαση των μπλοκ και ένα αρχείο με τον εκτελέσιμο κώδικα πίσω από κάθε μπλοκ.



Εικόνα 4-3. Περιβάλλον Block Exporter.

### 4.5 Εργαλείο *Workspace Factory*

Το Workspace Factory είναι ένα εργαλείο που παρέχει το Blockly για τη δημιουργία προσαρμοσμένων χώρων εργασίας (workspaces) με προκαθορισμένα μπλοκ, τα οποία μπορούν να χρησιμοποιηθούν σε εκπαιδευτικά ή προγραμματιστικά έργα. Μπορούν να περιοριστούν ή να επεκταθούν τα διαθέσιμα μπλοκ, ανάλογα με το επίπεδο ή τις ανάγκες του έργου. Επίσης μπορούν να δημιουργηθούν και να κατηγοριοποιηθούν τα μπλοκ με τρόπο που να είναι πιο εύκολα κατανοητά από τον χρήστη. Εκτός από το σύνολο των διαθέσιμων μπλοκ, μπορούν να καθοριστούν και άλλες παράμετροι, όπως τα εργαλεία που θα είναι ενεργοποιημένα, αν θα υπάρχουν προεπιλεγμένα μπλοκ στον χώρο εργασίας όταν ξεκινά ο χρήστης, και αν θα υπάρχει πρόσβαση σε συγκεκριμένα χαρακτηριστικά όπως εισαγωγή/εξαγωγή XML.







Εικόνα 4-4. Περιβάλλον Workspace Factory.

## 4.6 Ορισμός μπλοκ

Οι ορισμοί μπλοκ περιγράφουν πώς φαίνεται και συμπεριφέρεται ένα μπλοκ, συμπεριλαμβανομένου του κειμένου, του χρώματος, του σχήματος και με ποια άλλα μπλοκ μπορεί να συνδεθεί. Η τιμή χρώματος μπορεί να δοθεί σε μορφή HSV (Hue-Saturation-Value) ή Hex (Hexadecimal).

### 4.6.1 Συνδεσιμότητα

Οι χρήστες μπορούν να δημιουργήσουν ακολουθίες μπλοκ. Στην τυπική διάταξη του Blockly, αυτές οι συνδέσεις βρίσκονται στο επάνω και στο κάτω μέρος, με τα μπλοκ στοιβαγμένα κάθετα. Ένα μπλοκ με προηγούμενη υποδοχή δεν μπορεί να έχει υποδοχή εξόδου και το αντίστροφο. Επίσης όρος μπλοκ δηλώσεων αναφέρεται σε ένα μπλοκ χωρίς έξοδο τιμής. Τέλος, ένα μπλοκ δηλώσεων θα έχει συνήθως και μια προηγούμενη σύνδεση και μια επόμενη σύνδεση. Οι διαφορετικοί τρόποι σύνδεσης παρουσιάζονται στον Πίνακα 4.1.

Πίνακας 4-3. Τρόποι συνδεσιμότητας.	
Καμία σύνδεση με άλλα μπλοκ.	
Δυνατότητα σύνδεσης με προηγούμενο μπλοκ.	
Δυνατότητα σύνδεσης με επόμενο μπλοκ.	
Δυνατότητα σύνδεσης με προηγούμενο και επόμενο μπλοκ.	

#### 4.6.2 Τύποι μπλοκ

Ένα μπλοκ Εξόδου (Block Output) μπορεί να έχει μία μόνο έξοδο, που αναπαρίσταται ως σύνδεσμος αρσενικού παζλ στο μπροστινό άκρο. Οι εξοδοί συνδέονται με εισόδους τιμών. Τα μπλοκ με έξοδο ονομάζονται συνήθως μπλοκ τιμής (value blocks). Μπορεί να συνδεθεί με μπλοκ εισόδων και να αποτελεί το όρισμα αυτών. Ο χρήστης πρέπει να καθορίσει τον τύπο της εξόδου που θα επιστρέφει το μπλοκ από ένα σύνολο επιλογών (boolean, number, string, array κ.ά.).



Εικόνα 4-5. Μπλοκ Εξόδου.

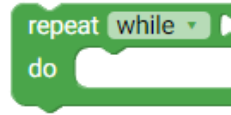
Ένα μπλοκ Εισόδου (Block Input) έχει μία ή περισσότερες εισόδους, όπου κάθε είσοδος έχει μια ακολουθία πεδίων και μπορεί να καταλήγει σε μια σύνδεση. Υπάρχουν διάφοροι τύποι ενσωματωμένων εισόδων. Υπάρχουν τέσσερις βασικοί τύποι Blocks Εισόδου, οι οποίοι διαφέρουν στον τύπο της συνδεσιμότητας με άλλα μπλοκ.

- Value input: Συνδέεται σε σύνδεση εξόδου ενός μπλοκ τιμών. Ένα μπλοκ «math\_arithmetic» (πρόσθεση, αφαίρεση) είναι ένα παράδειγμα μπλοκ με δύο εισόδους τιμών.



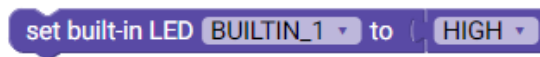
Εικόνα 4-6. Value input μπλοκ.

- **Statement input:** Συνδέεται σε προηγούμενη σύνδεση ενός μπλοκ δηλώσεων. Μπορεί να συνδεθεί με μπλοκ Εξόδου και εφόσον πληρείται κάποια συνθήκη να καλούνται εμφωλευμένα μπλοκ που βρίσκονται εντός του μπλοκ. Το ένθετο τμήμα ενός βρόχου while είναι ένα παράδειγμα.



Εικόνα 4-7. Statement input μπλοκ.

- **Dummy input:** Δεν έχει σύνδεση μπλοκ, δεν διαθέτει εγκοπή σύνδεσης στο δεξί μέρος για περαιτέρω σύνδεση. Λειτουργεί σαν νέα γραμμή όταν το μπλοκ έχει ρυθμιστεί να χρησιμοποιεί εξωτερικές εισόδους τιμών.



Εικόνα 4-8. Dummy input μπλοκ.

- **End-row input:** Δεν έχει σύνδεση μπλοκ και λειτουργεί πάντα σαν νέα γραμμή.



Εικόνα 4-9. End-row input μπλοκ.

## 4.7 Διαδικασία αρχικοποίησης για την ανάπτυξη του Ardublockly


Ο πηγαίος κώδικας για τη λήψη, περαιτέρω ανάπτυξη ή τροποποίηση της εφαρμογής Ardublockly είναι διαθέσιμος δωρεάν στο GitHub [24]. Στο παρόν σύστημα, η ανάπτυξη έγινε σε λειτουργικό Windows. Για να ξεκινήσει η διαδικασία ανάπτυξης του λογισμικού Ardublockly, θα πρέπει να ακολουθηθούν τα εξής βήματα:

**Βήμα 1:** Κατέβασμα απαιτούμενων εργαλείων για τη δημιουργία εκτελέσιμου.

Το *Git* πρέπει να είναι εγκατεστημένο στο σύστημα και προσβάσιμο μέσω της γραμμής εντολών. Στη συνέχεια, χρειάζονται *συγκεκριμένες εκδόσεις της Python*. Η "core έκδοση" του Ardublockly (διακομιστής γραμμής εντολών και γραφικό περιβάλλον που βασίζεται σε

πρόγραμμα περιήγησης) είναι πλήρως συμβατή τόσο με την Python 2 όσο και με την Python 3 (δοκιμασμένη σε Python 2.7 και 3.4), ωστόσο υπάρχει ένα συγκεκριμένο βήμα στη διαδικασία κατασκευής που απαιτεί Python 2.7. Επιπλέον, απαιτείται το *py2exe*, μια επέκταση για τη μετατροπή Python scripts σε εκτελέσιμα προγράμματα για Windows, καθώς και το *MkDocs*, ένα εργαλείο δημιουργίας στατικών σελίδων ειδικά σχεδιασμένο για τεκμηρίωση. Τέλος, το *Node.js* είναι απαραίτητο για την εκτέλεση του Electron και ο διαχειριστής πακέτων *npm* πρέπει να είναι επίσης εγκατεστημένος.

**Βήμα 2:** Κατέβασμα του πηγαίου κώδικα.

Ο ευκολότερος τρόπος για να κατέβει ένα πλήρες αντίγραφο από το αποθετήριο είναι να χρησιμοποιηθεί το git στο  Terminal :

```
git clone https://github.com/carlosperate/ardublockly.git
cd ardublockly
git submodule update --init --recursive
```

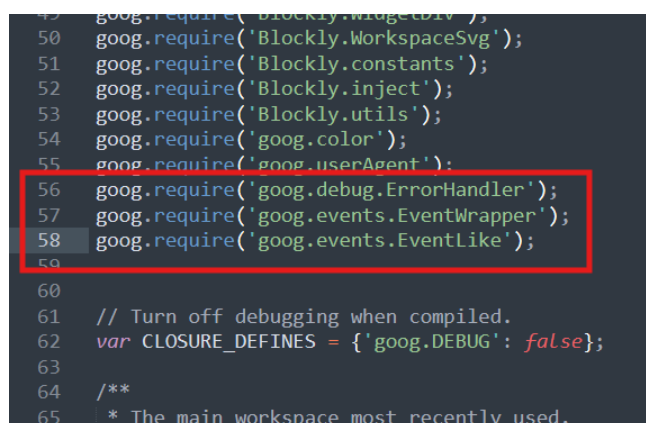
**Βήμα 3:** Κατέβασμα της βιβλιοθήκης closure library.

Το αρχείο closure-library-master.zip πρέπει να αποσυμπίεστεί μέσα στον βασικό φάκελο (root folder) ανάπτυξης της εφαρμογής και να μετονομαστεί σε closure-library.

**Βήμα 4:** Blockly Build.

Θα χρειαστεί η Python 2.7 για αυτό το βήμα, καθώς το σενάριο δημιουργίας Blockly δεν είναι προς το παρόν συμβατό με την Python 3.

Αρχικά στην διαδρομή: \ardublockly\blockly\core\blockly.js πρέπει να προστεθούν τρεις γραμμές κώδικα (Εικόνα 4.10).



```
49 goog.require('Blockly.WidgetDiv');
50 goog.require('Blockly.WorkspaceSvg');
51 goog.require('Blockly.constants');
52 goog.require('Blockly.inject');
53 goog.require('Blockly.utils');
54 goog.require('goog.color');
55 goog.require('goog.userAgent');
56 goog.require('goog.debug.ErrorHandler');
57 goog.require('goog.events.EventWrapper');
58 goog.require('goog.events.EventLike');
59
60
61 // Turn off debugging when compiled.
62 var CLOSURE_DEFINES = {'goog.DEBUG': false};
63
64 /**
65  * The main workspace most recently used.
```

Εικόνα 4-10. Προσθήκη κώδικα στο αρχείο blockly.js.

Έπειτα γίνεται εκτέλεση από τη γραμμή εντολών της εντολής: `python27 blockly\build.py`.

---



### **Βήμα 5:** Δημιουργία Python Server.

Για να διορθωθεί ένα συγκεκριμένο σφάλμα πρέπει να γίνει τροποποίηση του αρχείου `pyinstaller.spec`, το οποίο βρίσκεται στη διαδρομή `ardublockly\package`. Με τη χρήση κάποιου editor στη γραμμή 18, το `import ardublockly` αντικαθίσταται από το `import ardublocklyserver` (Εικόνα 4.11).

```

9  import sys
10 project_root = os.path.realpath('.')
11 print('PyInstaller defined project root: %s' % project_root)
12 local_packages = os.path.join(
13     project_root, 'ardublocklyserver', 'local-packages')
14 sys.path.insert(0, project_root)
15 sys.path.insert(0, local_packages)
16
17 # Import required modules, ensures PyInstaller fails if it cannot find them
18 import ardublocklyserver
19 import six, configparser, serial, waitress, bottle
20
21

```

Εικόνα 4-11. Αλλαγή κώδικα στο αρχείο `pyinstaller.spec`.

Για την δημιουργία του Python Server απαιτείται η εκτέλεση του `build_pyinstaller.py` εκτελώντας το από τον βασικό φάκελο : `python3.8 package\build_pyinstaller.py`.

### **Βήμα 6:** Δημιουργία της Εφαρμογής με το Electron.

Με την ολοκλήρωση του προηγούμενου βήματος δημιουργείται εντός του root folder ο φάκελος `arduexec` και το αρχείο `ardublockly_run.bat`. Το συγκεκριμένο αρχείο αποτελεί το εκτελέσιμο αρχείο για την εκκίνηση της εφαρμογής. Προκειμένου να ολοκληρωθεί η διαδικασία θα πρέπει να εκτελεστούν οι εξής εντολές:

```
cd package\electron
```

```
npm install
```

```
npm run release
```

Η διαδικασία που παρουσιάστηκε θα πρέπει να πραγματοποιηθεί μόνο μία φορά κατά τη φάση της αρχικοποίησης, για να στηθεί το περιβάλλον ανάπτυξης της εφαρμογής.

## **4.8 Διαδικασία ανάπτυξης του *Ardublockly***

Αφού ολοκληρωθεί επιτυχώς η διαδικασία προετοιμασίας, μπορεί να ξεκινήσει η ανάπτυξη της εφαρμογής. Αυτή η διαδικασία απαιτεί τη δημιουργία δύο αρχείων σε JavaScript (.js). Το πρώτο αρχείο περιλαμβάνει τον κώδικα που είναι υπεύθυνος για τη δημιουργία των μπλοκ (την εμφάνιση και τη λειτουργία τους μέσα στην εφαρμογή), ενώ το δεύτερο αρχείο περιέχει τον κώδικα που θα παράγεται όταν χρησιμοποιούνται τα μπλοκ στην εφαρμογή.

Επιπλέον, πρέπει να τροποποιηθούν δύο αρχεία, το `index.html` και το `ardublockly_toolbox.js`, τα οποία σχετίζονται με το γραφικό περιβάλλον της εφαρμογής και τη δημιουργία του μενού χρήστη. Στο πλαίσιο της ανάπτυξης του λογισμικού για την πλατφόρμα Ardublockly, δημιουργήθηκαν δύο αρχεία με το όνομα `my_block.js`, τα οποία πρέπει να έχουν το ίδιο όνομα.

Το αρχείο που περιέχει τον κώδικα για τη δημιουργία των μπλοκ (`my_block.js`), πρέπει να τοποθετηθεί στη διαδρομή `...\ardublockly\blockly\blocks\arduino`. Το αρχείο που περιέχει τον κώδικα που παράγεται από την χρήση των μπλοκ εντός της εφαρμογής (`my_block.js`), πρέπει να τοποθετηθεί στην διαδρομή `...\ardublockly\blockly\generators\arduino`. Ο κώδικας που περιέχεται στα αρχεία έχει προκύψει από ανάπτυξη κώδικα JavaScript που θα παρουσιαστεί παρακάτω. Σημειώνεται ότι:

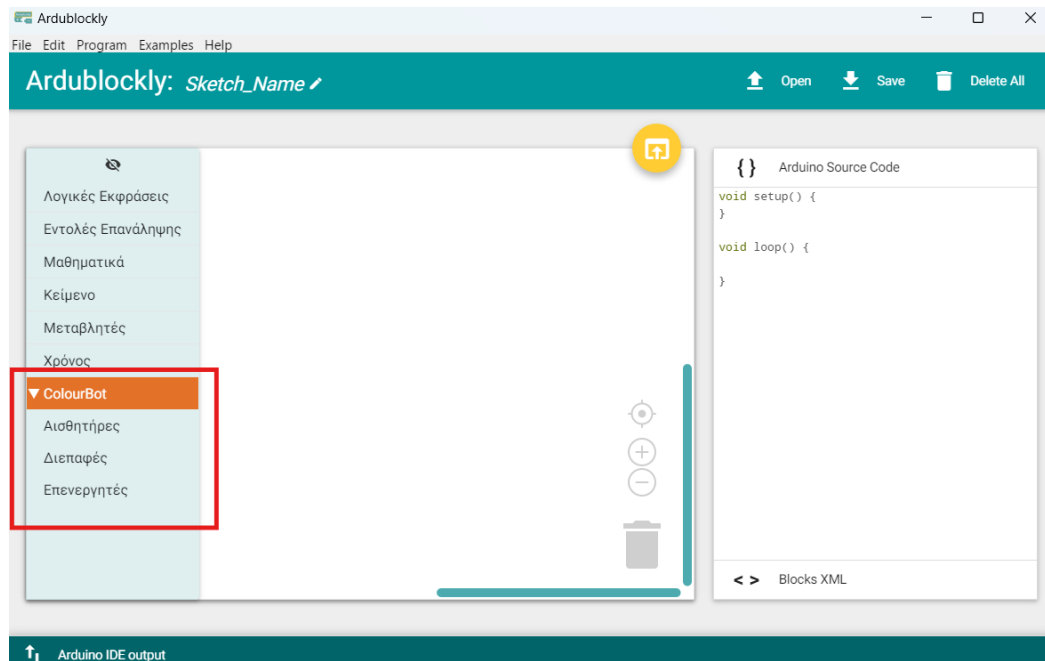
- Όταν ολοκληρωθεί η δημιουργία των δύο παραπάνω αρχείων, για να εφαρμοστούν οι αλλαγές στην εφαρμογή, θα πρέπει να γίνει χτίσιμο (build) της εφαρμογής. Αυτό πραγματοποιείται με την εκτέλεση του βήματος 4, όπως αναλύθηκε στην παράγραφο 4.8.
- Κάθε φορά που πραγματοποιείται οποιαδήποτε αλλαγή στα συγκεκριμένα αρχεία, θα πρέπει να εκτελείται το βήμα 4, για να εφαρμόζονται εκ νέου οι αλλαγές στην εφαρμογή.
- Συμπληρωματικά θα πρέπει να γίνεται καθαρισμός της προσωρινής μνήμης πριν από την εκτέλεση της διαδρομής, διαγράφοντας τον φάκελο `appdata`, που βρίσκεται στη διαδρομή: `...\ardublockly\arduexec`.
- Κάθε φορά που εκτελείται η εφαρμογή, ο φάκελος `appdata` δημιουργείται αυτόματα.

Το αρχείο `ardublockly_toolbox.js` είναι υπεύθυνο για τη δημιουργία του μενού χειρισμού (Εικόνα 4.12) των μπλοκ της εφαρμογής από τον χρήστη, έχει προστεθεί κώδικας (Εικόνα 4.13) ώστε να προσαρμοστεί στις ανάγκες του συστήματος. Το αρχείο περιέχει κώδικα σε γλώσσα XML και βρίσκεται στη διαδρομή: `..\ardublockly\ardublockly`. Οποιαδήποτε τροποποίηση του κώδικα που περιέχεται στο συγκεκριμένο αρχείο δεν απαιτεί εκ νέου χτίσιμο (build) της εφαρμογής. Για την εφαρμογή των αλλαγών χρειάζεται καθαρισμός της προσωρινής μνήμης.

Έπειτα, το αρχείο `index.html` είναι υπεύθυνο για τη δημιουργία του γραφικού περιβάλλοντος και του κύριου μενού της εφαρμογής. Ο κώδικας που περιέχει είναι γραμμένος σε γλώσσα HTML και το αρχείο βρίσκεται στη διαδρομή: `...\ardublockly\ardublockly`.

Πολλά από τα μπλοκ της εφαρμογής περιέχουν εικονίδια. Για την ενσωμάτωση εικόνας εντός του μπλοκ, θα πρέπει τα εικονίδια να βρίσκονται εντός της συγκεκριμένης διαδρομής:

..\ardublockly\ardublockly\img. Οποιαδήποτε τροποποίηση σε αρχείο εικόνας ή νέα ενσωμάτωση εικόνας σε μπλοκ, επιβάλλει εκ νέου χτίσιμο (build) της εφαρμογής.



Εικόνα 4-12. Μενού χρήστη.

```
' <sep></sep>' +
' <category id="catMyBlock" name="ColourBot">' +

' <category name="Αισθητήρες">' +
' <block type="my_block_yes_no"></block>' +
' <block type="my_block_ultrasonic"></block>' +
' <block type="my_block_distance_variable"></block>' +
' <block type="my_block_humidity_temperature"></block>' +
' <block type="my_block_temperature_variable"></block>' +
' <block type="my_block_humidity_variable"></block>' +
' <block type="my_block_infrared"></block>' +
' <block type="my_block_obstacle_variable"></block>' +
' <block type="my_block_line"></block>' +
' <block type="my_block_line_variable"></block>' +
' </category>' +

' <category name="Διεπαφές">' +
' <block type="my_block_highlow"></block>' +
' <block type="my_block_yes_no"></block>' +
' <block type="my_block_buttons"></block>' +
' <block type="my_block_button1"></block>' +
' <block type="my_block_button2"></block>' +
' <block type="my_block_button3"></block>' +
' <block type="my_block_3leds"></block>' +
' <block type="my_block_I2C_lcd"></block>' +
' <block type="my_block_joystick"></block>' +
' <block type="my_block_frontback_variable"></block>' +
' </category>' +

' <category name="Επενεργητές">' +
' <block type="my_block_motor">' +
' <value name="pwm_speed">' +
' <block type="math_number">' +
' <field name="NUM">0</field>' +
' </block>' +
' </value>' +
' </block>' +
' <block type="my_block_servo">' +
' <value name="rotation_to">' +
' <block type="math_number">' +
' <field name="NUM">0</field>' +
' </block>' +
' </value>' +
' </block>' +
' </category>' +
```

Εικόνα 4-13. Κώδικας XML που προστέθηκε στο αρχείο ardublockly\_toolbox.js.

Επιπλέον, κάποιες προσθήκες πρέπει να υλοποιηθούν για να εμφανιστεί η κατηγορία που δημιουργήθηκε. Πρώτον, στο αρχείο ardublockly\_blockly.js στην διαδρομή ...\\ardublockly\\ardublockly\\ ardublockly\_blockly.js γίνεται μία προσθήκη στην σελίδα 216 (Εικόνα 4.14).

```
212 /** Update the toolbox categories language. */
213 Ardublockly.updateToolboxLanguage = function() {
214   var categories = ['catLogic', 'catLoops', 'catMath', 'catText',
215                     'catVariables', 'catFunctions', 'catInputOutput',
216                     'catTime', 'catAudio', 'catMotors', 'catComms', 'catMyBlock'];
217   var categoryNodes = Ardublockly.xmlTree.getElementsByTagName('category');
218   for (var i = 0, cat; cat = categoryNodes[i]; i++) {
219     var catId = cat.getAttribute('id');
```

Εικόνα 4-14. Κώδικας που προστέθηκε στο αρχείο ardublockly\_blockly.js.

Δεύτερον, στο αρχείο index.html στην διαδρομή ...\\ardublockly\\ardublockly\\index.html πρέπει να προστεθούν δυο γραμμές κώδικα (Εικόνα 4.15):

```

34 <script src="../../blockly/blocks/arduino/spi.js"></script>
35 <script src="../../blockly/blocks/arduino/stepper.js"></script>
36 <script src="../../blockly/blocks/arduino/time.js"></script>
37 <script src="../../blockly/blocks/arduino/tone.js"></script>
38 <script src="../../blockly/blocks/arduino/variables.js"></script>
39 <script src="../../blockly/blocks/arduino/my_block.js"></script>
40 <script src="../../blockly/generators/arduino.js"></script>
41 <script src="../../blockly/generators/arduino/boards.js"></script>
42 <script src="../../blockly/generators/arduino/io.js"></script>
43 <script src="../../blockly/generators/arduino/lists.js"></script>
44 <script src="../../blockly/generators/arduino/logic.js"></script>
45 <script src="../../blockly/generators/arduino/loops.js"></script>
46 <script src="../../blockly/generators/arduino/map.js"></script>
47 <script src="../../blockly/generators/arduino/math.js"></script>
48 <script src="../../blockly/generators/arduino/procedures.js"></script>
49 <script src="../../blockly/generators/arduino/serial.js"></script>
50 <script src="../../blockly/generators/arduino/servo.js"></script>
51 <script src="../../blockly/generators/arduino/spi.js"></script>
52 <script src="../../blockly/generators/arduino/stepper.js"></script>
53 <script src="../../blockly/generators/arduino/text.js"></script>
54 <script src="../../blockly/generators/arduino/time.js"></script>
55 <script src="../../blockly/generators/arduino/tone.js"></script>
56 <script src="../../blockly/generators/arduino/variables.js"></script>
57 <script src="../../blockly/generators/arduino/my_block.js"></script>
58

```

Εικόνα 4-15. Κώδικας που προστέθηκε στο αρχείο index.html.

Και τρίτο, στο αρχείο en.js ή el.js (ανάλογα την γλώσσα εμφάνισης που έχει επιλεγεί) προστίθεται μια γραμμή κώδικα (Εικόνα 4.16).

```

87 catMath: "Μαθηματικά",
88 catText: "Κείμενο",
89 catVariables: "Μεταβλητές",
90 catFunctions: "Functions",
91 catInputOutput: "Input/Output",
92 catTime: "Χρόνος",
93 catAudio: "Audio",
94 catMotors: "Motors",
95 catComms: "Comms",
96 catMyBlock: "ColourBot",
97 },
98

```

Εικόνα 4-9. Κώδικας που προστέθηκε στο αρχείο en.js ή el.js.

## Βήμα 7: Μετατροπή της εφαρμογής σε εκτελέσιμη μορφή (.exe).

Μετά την ολοκλήρωση των βημάτων της παραγράφου 4.8 και την ανάπτυξη της εφαρμογής, το τελευταίο βήμα είναι η μετατροπή της εφαρμογής σε εκτελέσιμη μορφή (.exe). Αν και προαιρετικό για τη λειτουργία της, συνιστάται να πραγματοποιηθεί, καθώς καθιστά την εφαρμογή ανεξάρτητη από την εγκατάσταση της Python στον υπολογιστή του χρήστη. Αν δεν γίνει αυτό το βήμα, θα χρειαστεί να εγκατασταθεί η Python στον υπολογιστή. Επιπλέον, για να ανοίξει η εφαρμογή, εκτός από την εκτέλεση του αρχείου ardublockly\_run.bat, θα πρέπει κάθε φορά να εκτελείται και το αρχείο start.py, το οποίο ενεργοποιεί την επικοινωνία της εφαρμογής Ardublockly με το Arduino IDE. Συνιστάται να ολοκληρωθεί αυτό το βήμα για πλήρη

ανάπτυξη της εφαρμογής. Για να δημιουργηθεί η εκτελέσιμη μορφή του λογισμικού, εκτελείται στη γραμμή εντολών στο βασικό φάκελο η εντολή: `..python package/pack_ardublockly.py`.

## ΚΕΦΑΛΑΙΟ 5: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

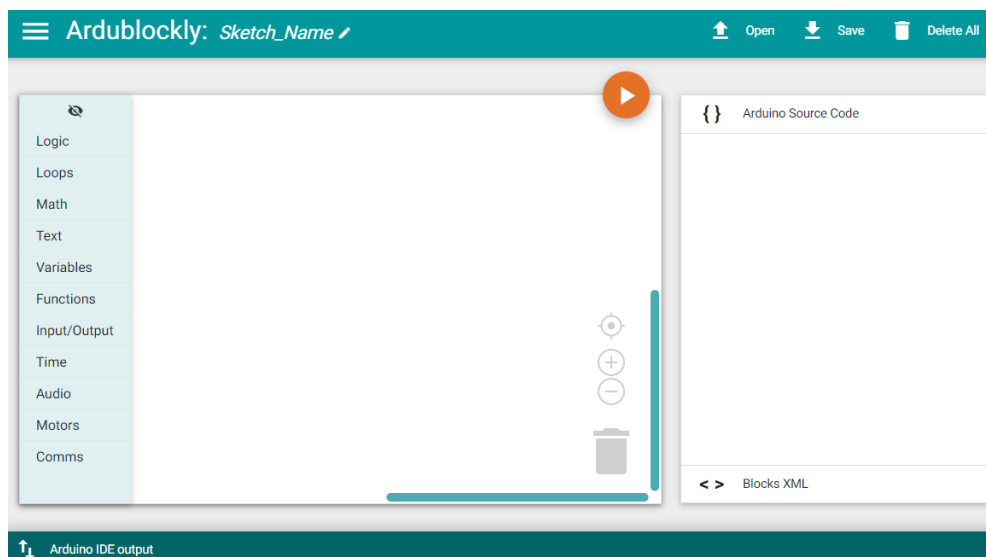
---

Σε αυτό το κεφάλαιο παρουσιάζεται το λογισμικό που αναπτύχθηκε για να καταστεί λειτουργική η πλατφόρμα και τα αρθρώματα της, όπως αναφέρθηκαν στο Κεφάλαιο 3. Το λογισμικό αυτό βασίζεται στο Ardublockly, το οποίο έχει τροποποιηθεί και εμπλουτιστεί με τα κατάλληλα δομικά στοιχεία (μπλοκ), ώστε να μπορεί να αλληλοεπιδρά με τα σχεδιασμένα αρθρώματα, καθιστώντας τα πλήρως αξιοποιήσιμα από τους χρήστες. Βασική προτεραιότητα αποτέλεσε η προσαρμογή του λογισμικού στις ανάγκες και την ηλικία των χρηστών. Έχει αναπτυχθεί μια ειδική έκδοση για μαθητές δημοτικού. Αυτή περιλαμβάνει απλά μπλοκ, και ο χρήστης πρέπει να επιλέξει απλές παραμέτρους, όπως άνοιγμα/κλείσιμο ή μπροστά/πίσω, καθώς και τον αριθμό της θύρας στην οποία έχει ήδη συνδέσει το καλώδιο οποιουδήποτε αρθρώματος.

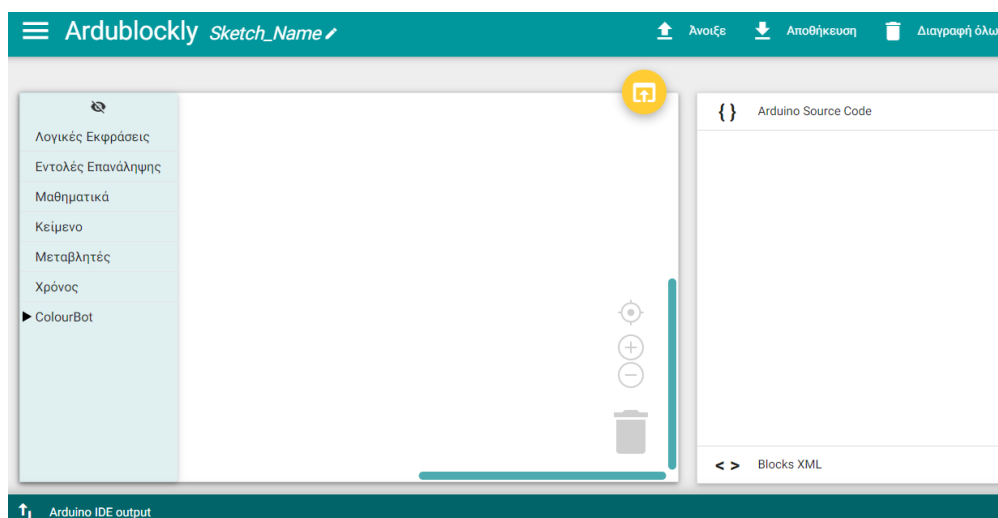
### ***5.1 Προσαρμογή του λογισμικού Ardublockly στις ανάγκες της εκπαιδευτικής πλατφόρμας***

Η εφαρμογή αναβαθμίστηκε σημαντικά με την προσθήκη νέων μπλοκ, τα οποία σχεδιάστηκαν ειδικά για να αξιοποιούν πλήρως τις δυνατότητες του υλικού που παρουσιάστηκε σε προηγούμενο κεφάλαιο. Αυτή η διαδικασία προσαρμογής δεν έγινε επιφανειακά. Προηγήθηκε εκτενής ανάλυση και μελέτη της αρχικής μορφής της εφαρμογής (Εικόνα 5.1), ώστε να διασφαλιστεί ότι η νέα σχεδίαση θα προσφέρει στους χρήστες, κυρίως μαθητές Πρωτοβάθμιας εκπαίδευσης, μια φιλική και εύχρηστη εμπειρία. Αφαιρέθηκαν ορισμένα μπλοκ που κρίθηκαν περιττά ή πολύπλοκα για τους συγκεκριμένους χρήστες, ενώ το μενού επανασχεδιάστηκε με γνώμονα τη λειτουργικότητα και την απλότητα. Παράλληλα, προστέθηκαν νέα μπλοκ, διευκολύνοντας τους χρήστες να αλληλεπιδράσουν με το σύστημα με ευκολία και αποτελεσματικότητα. Η αναβαθμισμένη έκδοση της εφαρμογής, εμφανίζει ένα πιο

φιλικό μενού για τους μαθητές, όπως παρουσιάζεται στην Εικόνα 5.2, κάνοντας τη χρήση της εφαρμογής πιο άμεση και προσιτή. Η νέα εφαρμογή τροποποιήθηκε στα Ελληνικά.



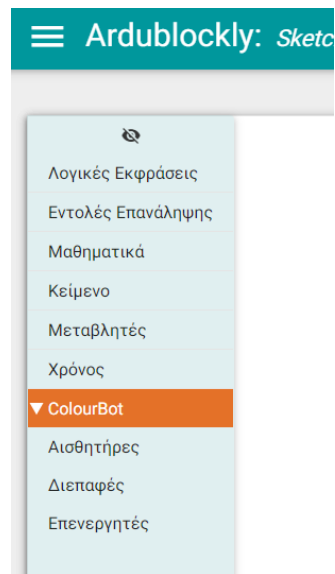
Εικόνα 5-1. Αρχικό μενού της εφαρμογής.



Εικόνα 5-2. Τροποποιημένο μενού της εφαρμογής.

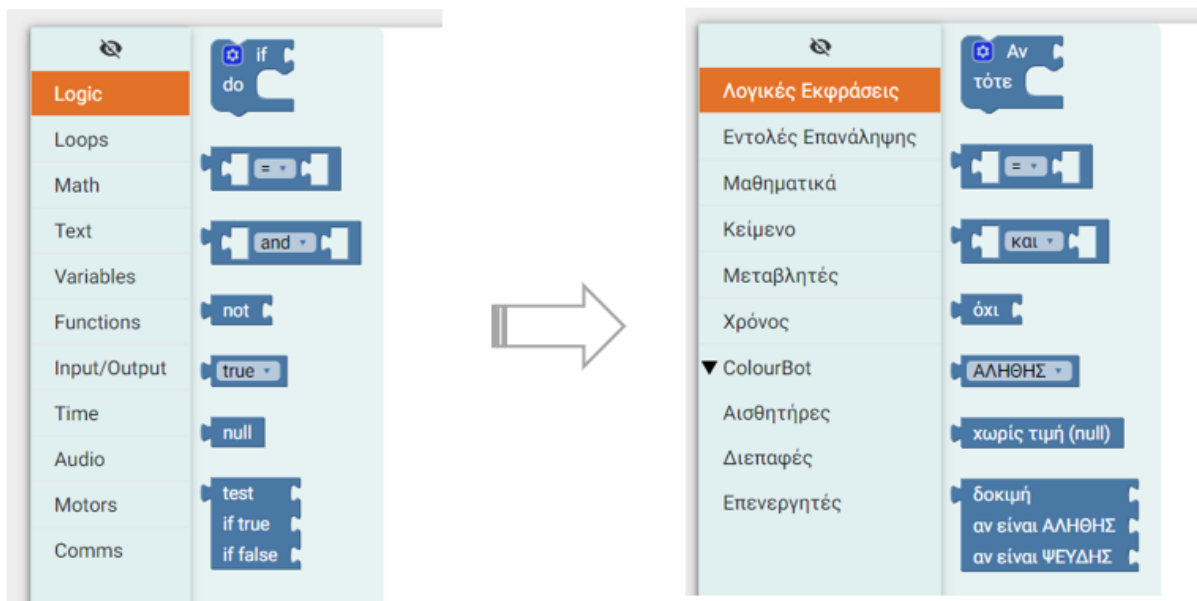
Στα πλαίσια της προσαρμογής του Ardublockly στην νέα έκδοση έγιναν αρκετές τροποποιήσεις στις περισσότερες κατηγορίες. Για τη λειτουργία των επιμέρους μπλοκ που υπάρχουν και στην αρχική έκδοση, ο αναγνώστης μπορεί να ανατρέξει στις οδηγίες χρήσης της πλατφόρμας Ardublockly [25]. Από το μενού χρήστη μπορεί να επιλεγεί η επιθυμητή κατηγορία (Εικόνα 5.3). Στη συνέχεια εμφανίζονται οι επιλογές της κατηγορίας που επιλέχθηκε, οι οποίες είναι τα μπλοκ όπου θα χρησιμοποιηθούν.





Εικόνα 5-3. Αρχικό μενού.

Όσον αφορά τις κατηγορίες Logic, Loops, Math, Text, Variables, Time δεν αφαιρέθηκε κάποιο από τα αρχικά μπλοκ αλλά μεταφράστηκαν στα ελληνικά. Ενδεικτικά στην Εικόνα 5.4 παρουσιάζεται η κατηγορία Logic μεταφρασμένη.



Εικόνα 5-4. Κατηγορία Logic.

Οι κατηγορίες Functions, Input/Output, Audio, Motors δεν συμπεριλήφθηκαν. Οι Functions και Audio θεωρήθηκαν περιττές και οι υπόλοιπες καλύφθηκαν από τα μπλοκ που δημιουργήθηκαν στην κατηγορία ColourBot.

## 5.2 Μπλοκ της κατηγορίας *ColourBot*

Στην συγκεκριμένη ενότητα παρουσιάζονται τα μπλοκ που έχουν δημιουργηθεί για τις ανάγκες της παρούσας εργασίας, ο κώδικας που παράχθηκε για την δημιουργία τους και ο τρόπος λειτουργίας τους. Τα μπλοκ χωρίζονται σε τρεις κατηγορίες: α) Αισθητήρες, β) Διεπαφές και γ) Επενεργητές.

Λόγο της ηλικίας των χρηστών δεν απαιτείται η επιλογή των ακροδεκτών από αυτούς. Υπάρχει χρωματικός κώδικας, και με βάση την σχεδίαση του υλικού και των αρθρωμάτων, έχουν οριστεί αριθμητικές χρωματικές θύρες που οι ίδιες καθορίζουν τα pin σύνδεσης στον Arduino.

### Αισθητήρες

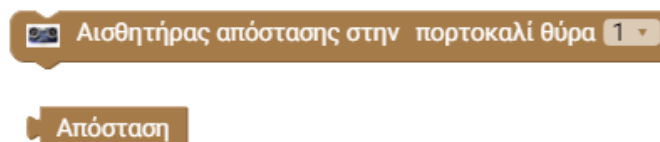
#### 5.2.1 Μπλοκ *Sonar Sensor*

Στην Εικόνα 5.5 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ και στην Εικόνα 5.6 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_ultrasonic'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(32);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/ultrasonic.png", 20, 20, "*"))
      .appendField("Αισθητήρας απόστασης στην")
      .appendField(new Blockly.FieldDropdown([["πορτοκαλί θύρα 1", "or1"], ["πορτοκαλί θύρα 2", "or2"]], 'orange_ports'));
    this.setTooltip("Το αποτέλεσμα του αισθητήρα είναι η Απόσταση σε cm από ένα εμπόδιο. Βλέπει από 2cm έως 500cm.");
  }
};

Blockly.Blocks['my_block_distance_variable'] = {
  init: function() {
    this.setColour(32);
    this.setOutput(true, Blockly.Types.NUMBER.output);
    this.appendDummyInput()
      .appendField("Απόσταση")
    this.setTooltip("Αποτέλεσμα αισθητήρα απόστασης σε cm.");
  },
  getBlockType: function() {
    return Blockly.Types.NUMBER;
  }
};
```

Εικόνα 5-5. Κώδικας JavaScript για την δημιουργία του μπλοκ Sonar Sensor.



Εικόνα 5-6. Εικονικό μπλοκ Sonar Sensor.

Για την χρήση του αισθητήρα υπερήχων στο λογισμικό είναι απαραίτητη η χρήση και των δύο εικονικών μπλοκ της Εικόνας 5.6. Όπως φαίνεται στην Εικόνα 5.8 με τη χρήση του πρώτου μπλοκ αρχικά ορίζεται το μέγιστο εύρος μέτρησης στα 520, η ανάλυση του αναλογικού-ψηφιακού μετατροπέα στα 1023 και καθορίζεται ο ακροδέκτης σύνδεσης. Στη συνάρτηση `loop()`, η τιμή από τον αισθητήρα διαβάζεται μέσω της `analogRead()` και αποθηκεύεται σε μια μεταβλητή. Στη συνέχεια, η απόσταση υπολογίζεται μέσω μιας εξίσωσης μετατρέποντας την αναλογική τιμή σε πραγματική απόσταση. Το μπλοκ Απόσταση επιστρέφει την απόσταση του εμποδίου από τον αισθητήρα σε cm.

Στην Εικόνα 5.7 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.8 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_ultrasonic'] = function(block) {
  var orange_ports = block.getFieldValue('orange_ports');

  if (orange_ports == 'or1'){
    var pin = 'A0';
  }
  else if(orange_ports == 'or2'){
    var pin = 'A6';
  };

  var ultrasonicName = 'ultrasonic';

  Blockly.Arduino.addDeclaration('ultrasonic_' + pin , '//Define ultrasonic\n'+ '#define MAX_RANG (520)\n'
    + '#define ADC_SOLUTION (1023.0)\n'+ 'int sensityPin = ' + pin + ' ; // select the input pin\n' + 'float dist_t, sensity_t;\n');

  var code1='sensity_t = analogRead(sensityPin);\n';
  var code2='dist_t = sensity_t * MAX_RANG / ADC_SOLUTION;\n';
  var code =code1+code2;
  return code;
};

Blockly.Arduino['my_block_distance_variable'] = function(block) {
  var code = 'dist_t';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-7. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Sonar Sensor.

```
//Define ultrasonic
#define MAX_RANG (520)
#define ADC_SOLUTION (1023.0)
int sensityPin =A0; // select the input pin
float dist_t, sensity_t;

void setup() {
}

void loop() {
  sensity_t = analogRead(sensityPin);
  dist_t = sensity_t * MAX_RANG / ADC_SOLUTION;
}
```

Εικόνα 5-8. Παραγόμενος κώδικας C++ μπλοκ Sonar Sensor.

### 5.2.2 Μπλοκ Temperature & Humidity

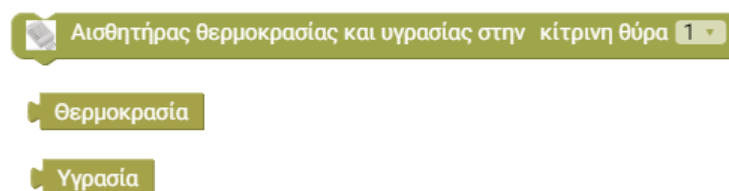
Στην Εικόνα 5.9 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ και στην Εικόνα 5.10 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_humidity_temperature'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(64);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/hum_tem.png", 20, 20, "*"));
    this.appendField("Αισθητήρας θερμοκρασίας και υγρασίας στην");
    this.appendField(new Blockly.FieldDropdown([["κίτρινη θύρα 1","ye1"], ["κίτρινη θύρα 2","ye2"], ["κίτρινη θύρα 3","gr3"]]), 'yellow_ports');
    this.setTooltip("Όρισε σε ποιά κίτρινη θύρα συνδέετε ο αισθητήρας. Το αποτέλεσμα του είναι η θερμοκρασία και η υγρασία.");
  }
};

Blockly.Blocks['my_block_temperature_variable'] = {
  init: function() {
    this.setColour(58);
    this.setOutput(true, Blockly.Types.NUMBER.output);
    this.appendDummyInput()
      .appendField("Θερμοκρασία");
    this.setTooltip("Αποτέλεσμα αισθητήρα θερμοκρασίας και Υγρασίας.");
  },
  getBlockType: function() {
    return Blockly.Types.NUMBER;
  }
};

Blockly.Blocks['my_block_humidity_variable'] = {
  init: function() {
    this.setColour(58);
    this.setOutput(true, Blockly.Types.NUMBER.output);
    this.appendDummyInput()
      .appendField("Υγρασία");
    this.setTooltip("Αποτέλεσμα αισθητήρα θερμοκρασίας και Υγρασίας.");
  },
  getBlockType: function() {
    return Blockly.Types.NUMBER;
  }
};
```

Εικόνα 5-9. Κώδικας JavaScript για την δημιουργία του μπλοκ Temperature & Humidity.



Εικόνα 5-10. Εικονικό μπλοκ Temperature & Humidity.

Για την χρήση του αισθητήρα θερμοκρασίας και υγρασίας στο λογισμικό είναι απαραίτητη η χρήση και των τριών εικονικών μπλοκ της Εικόνας 5.10. Όπως παρουσιάζεται στην Εικόνα 5.12 με τη χρήση του πρώτου μπλοκ, αρχικά γίνεται εισαγωγή της βιβλιοθήκης DHT.h, που περιέχει τις απαραίτητες συναρτήσεις για τη λειτουργία του αισθητήρα. Στη συνέχεια, ορίζεται ο ακροδέκτης στον οποίο συνδέεται ο αισθητήρας καθώς και ο τύπος του αισθητήρα. Στη συνάρτηση setup(), ξεκινάει η επικοινωνία με τον αισθητήρα και μέσα στη συνάρτηση loop() ο κώδικας διαβάζει τις τιμές της υγρασίας και της θερμοκρασίας από τον

αισθητήρα κάθε 2 δευτερόλεπτα (2000ms καθυστέρηση). Οι τιμές αποθηκεύονται στις μεταβλητές hum (υγρασία) και temp (θερμοκρασία). Το δεύτερο και τρίτο μπλοκ επιστρέφουν την θερμοκρασία και υγρασία που ανιχνεύτηκε.

Στην Εικόνα 5.11 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.12 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_humidity_temperature'] = function(block) {
  var yellow_ports = block.getFieldValue('yellow_ports');

  if (yellow_ports == 'ye1'){
    var pin = '11';
  }
  else if(yellow_ports == 'ye2'){
    var pin = '10';
  }
  else{
    var pin = '9';
  };

  var dhtName = 'dht';

  Blockly.Arduino.addInclude('dht', '//Library for DHT-22 sensor\n'+ '#include "DHT.h"');
  Blockly.Arduino.addDeclaration('my_block_' + pin, '//Defines for dht\n'+ '#define DHTPIN ' + pin + '\n' + '#define DHTTYPE DHT22\n'
    + 'DHT ' + dhtName + '(DHTPIN, DHTTYPE);\n');

  var pinSetupCode = dhtName + '.begin();'
  Blockly.Arduino.addSetup(dhtName, pinSetupCode, true);

  var code1 = '// Wait a few seconds between measurements.\ndelay(2000);\n';
  var code2 = 'float hum = dht.readHumidity();\n';
  var code3 = 'float temp = dht.readTemperature();\n';
  var code=code1+code2+code3;
  return code;
};

Blockly.Arduino['my_block_temperature_variable'] = function(block) {
  var code = 'temp';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};

Blockly.Arduino['my_block_humidity_variable'] = function(block) {
  var code = 'hum';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-11. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Temperature & Humidity.

```
//Library for DHT-22 sensor
#include "DHT.h"

//Defines for dht
#define DHTPIN 11
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);
  float hum = dht.readHumidity();
  float temp = dht.readTemperature();
}
```

Εικόνα 5-12. Παραγόμενος κώδικας C++ μπλοκ Temperature & Humidity.

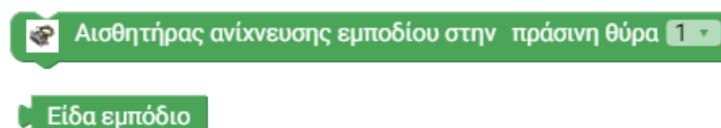
### 5.2.3 Μπλοκ Infrared Sensor

Στην Εικόνα 5.13 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ και στην Εικόνα 5.14 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_infrared'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/infrared.jpg", 20, 20, "*"))
      .appendField("Αισθητήρας ανίχνευσης εμποδίου στην")
      .appendField(new Blockly.FieldDropdown([["πράσινη θύρα 1", "gr1"], ["πράσινη θύρα 2", "gr2"]]), 'green_ports');
    this.setColour(130);
    this.setTooltip("Όρισε σε ποια πράσινη θύρα σύνδεσες τον αισθητήρα και δεξ αν εντοπίζει εμπόδιο. Το αποτέλεσμα του αισθητήρα είναι το Είδα εμπόδιο");
  }
};

Blockly.Blocks['my_block_obstacle_variable'] = {
  init: function() {
    this.setColour(130);
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.appendDummyInput()
      .appendField("Είδα εμπόδιο")
      .setTooltip("Μεταβλητή ανίχνευσης εμποδίου");
  },
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};
```

Εικόνα 5-13. Κώδικας JavaScript για την δημιουργία του μπλοκ Infrared Sensor.



Εικόνα 5-14. Εικονικό μπλοκ Infrared Sensor.

Για τη χρήση του αισθητήρα υπέρυθρων στο λογισμικό είναι απαραίτητη η χρήση και των δύο εικονικών μπλοκ της Εικόνας 5.14. Με τη χρήση του πρώτου μπλοκ όπως παρουσιάζεται και στην Εικόνα 5.16 αρχικά γίνεται αρχικοποίηση του ακροδέκτη που είναι συνδεδεμένος ο αισθητήρας (ορίζεται ως είσοδος με τη χρήση της `pinMode()`). Στη συνέχεια, στη συνάρτηση `loop()`, γίνεται ανάγνωση της ψηφιακής κατάστασης του αισθητήρα. Η τιμή αποθηκεύεται στη μεταβλητή `obstacle`, η οποία θα είναι είτε 1 (HIGH) αν ανιχνεύεται εμπόδιο, είτε 0 (LOW) αν δεν ανιχνεύεται τίποτα. Το δεύτερο μπλοκ επιστρέφει την τιμή της μεταβλητής για χρήση σε κάποια συνθήκη.

Στην Εικόνα 5.15 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.16 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_infrared'] = function(block) {
  var green_ports = block.getFieldValue('green_ports');

  if (green_ports == 'gr1'){
    var pin = '17';
  }
  else{
    var pin = '16';
  };

  var pinSetupCode = 'pinMode(' + pin + ', INPUT);';
  Blockly.Arduino.addSetup('my_block_' + pin, pinSetupCode, false);

  var code='int obstacle = digitalRead(' + pin + ');\\n';
  return code;
};

Blockly.Arduino['my_block_obstacle_variable'] = function(block) {
  var code = 'obstacle';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-15. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Infrared Sensor.

```
void setup() {
  pinMode(17, INPUT);
}

void loop() {
  int obstacle = digitalRead(17);
}
```

Εικόνα 5-16. Παραγόμενος κώδικας C++ μπλοκ Infrared Sensor.

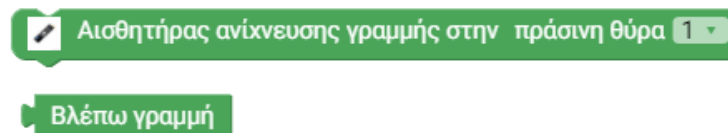
### 5.2.4 Μπλοκ Line Tracking

Στην Εικόνα 5.17 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ και στην Εικόνα 5.18 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_line'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/line.jpg", 20, 20, ""))
      .appendField("Αισθητήρας ανίχνευσης γραμμής στην")
      .appendField(new Blockly.FieldDropdown([["πράσινη θύρα 1", "gr1"], ["πράσινη θύρα 2", "gr2"]]), 'green_ports');
    this.setColour(130);
    this.setTooltip("Όρισε σε ποια πράσινη θύρα σύνδεσες τον αισθητήρα και δες αν βλέπει γραμμή. Το αποτέλεσμα του αισθητήρα είναι το βλέπω γραμμή");
  }
};

Blockly.Blocks['my_block_line_variable'] = {
  init: function() {
    this.setColour(130);
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.appendDummyInput()
      .appendField("Βλέπω γραμμή");
    this.setTooltip("Μεταβλητή ανίχνευσης γραμμής");
  },
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};
```

Εικόνα 5-17. Κώδικας JavaScript για την δημιουργία του μπλοκ Line Tracking.



Εικόνα 5-18. Εικονικό μπλοκ Line Tracking.

Για την χρήση του αισθητήρα υπέρυθρων για την ανίχνευση διαδρομής, στο λογισμικό είναι απαραίτητη η χρήση και των δύο εικονικών μπλοκ της Εικόνας 5.18. Με τη χρήση του πρώτου μπλοκ αρχικά γίνεται αρχικοποίηση του ακροδέκτη που είναι συνδεδεμένος ο αισθητήρας όπως φαίνεται και στην Εικόνα 5.20 (ορίζεται ως είσοδος με τη χρήση της pinMode()). Στη συνέχεια, στη συνάρτηση loop(), γίνεται ανάγνωση της ψηφιακής κατάστασης του αισθητήρα. Η τιμή αποθηκεύεται στη μεταβλητή line, η οποία θα είναι είτε 1 (HIGH) αν ανιχνεύεται γραμμή, είτε 0 (LOW) αν δεν ανιχνεύεται τίποτα. Το δεύτερο μπλοκ επιστρέφει την τιμή της μεταβλητής για χρήση σε κάποια συνθήκη.

Στην Εικόνα 5.19 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.20 όπου είναι και ο κώδικας που βλέπει ο χρήστης.



```
Blockly.Arduino['my_block_line'] = function(block) {
  var green_ports = block.getFieldValue('green_ports');

  if (green_ports == 'gr1'){
    var pin = '17';
  }
  else{
    var pin = '16';
  };

  var pinSetupCode = 'pinMode(' + pin + ', INPUT);';
  Blockly.Arduino.addSetup('my_block_' + pin, pinSetupCode, false);

  var code='int line = digitalRead(' + pin + ');\\n';
  return code;
};

Blockly.Arduino['my_block_line_variable'] = function(block) {
  var code = 'line';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-19. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Line Tracking.

```
void setup() {
  pinMode(17, INPUT);
}

void loop() {
  int line = digitalRead(17);
}
```

Εικόνα 5-20. Παραγόμενος κώδικας C++ μπλοκ Line Tracking.

## Διεπαφές

### 5.2.5 Μπλοκ LED

Στις Εικόνες 5.21 και 5.22 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ των LED αλλά και του μπλοκ που το συνοδεύει ενώ στην Εικόνα 5.23 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_highlow'] = {
  init: function() {
    this.setColour(178);
    this.appendDummyInput()
      .appendField(new Blockly.FieldDropdown([["ΑΝΟΙΧΤΟ", 'HIGH'], ["ΚΛΕΙΣΤΟ", 'LOW']], 'STATE'));
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.setTooltip("Λογική τιμή ΑΝΟΙΧΤΟ = HIGH(1) ή ΚΛΕΙΣΤΟ = LOW(0)");
  },
  /** @return {!string} The type of return value for the block, an integer. */
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};
```

Εικόνα 5-21. Κώδικας JavaScript για την δημιουργία του μπλοκ Ανοιχτό/Κλειστό.

```
Blockly.Blocks['my_block_3leds'] = {
  init: function() {
    this.setColour(0);
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/leds.jpg", 20, 20, "*"))
      .appendField("Φωτάκια LED στην")
      .appendField(new Blockly.FieldDropdown([["κόκκινη θύρα 1", "red1"], ["κόκκινη θύρα 2", "red2"]]), 'red_ports');
    this.appendValueInput('STATE1')
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendField(new Blockly.FieldImage("img/ColourBot/red-circle.jpg", 20, 20, "*"))
      .appendField("Κόκκινο Led ")
      .setCheck(Blockly.Types.BOOLEAN.checkList);
    this.appendValueInput('STATE2')
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendField(new Blockly.FieldImage("img/ColourBot/yellow-circle.jpg", 20, 20, "*"))
      .appendField("Κίτρινο Led ")
      .setCheck(Blockly.Types.BOOLEAN.checkList);
    this.appendValueInput('STATE3')
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendField(new Blockly.FieldImage("img/ColourBot/green-circle.jpg", 20, 20, "*"))
      .appendField("Πράσινο Led ")
      .setCheck(Blockly.Types.BOOLEAN.checkList);
    this.setTooltip("Δώσε τιμή ΑΝΟΙΧΤΟ ή ΚΛΕΙΣΤΟ στα φωτάκια LED ανάλογα πώς θέλεις να ανάψουν.");
  }
};
```

Εικόνα 5-22. Κώδικας JavaScript για την δημιουργία του μπλοκ LED.



Εικόνα 5-23. Εικονικό μπλοκ LED και εικονικό μπλοκ Ανοιχτό/Κλειστό.

Το μπλοκ LED παρουσιάζεται στην Εικόνα 5.23. Μαζί του υπάρχει η επιλογή Ανοιχτό/Κλειστό σαν μπλοκ για να αλλάζει η κατάσταση των ψηφιακών σημάτων σε HIGH/LOW. Με τη χρήση του μπλοκ ελέγχονται τρεις ψηφιακοί ακροδέκτες οι οποίοι έχουν οριστεί ως έξοδοι προετοιμάζοντας το σύστημα για τον έλεγχο των LED. Στη συνάρτηση loop() όπως φαίνεται στην Εικόνα 5.26, όλες οι έξοδοι αυτές τίθενται στην κατάσταση LOW, δηλαδή απενεργοποιούνται (μηδενική τάση).

Στις Εικόνες 5.24 και 5.25 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθούν τα μπλοκ. Το αποτέλεσμα τους φαίνεται στην Εικόνα 5.26 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_3leds'] = function(block) {
  var red_ports = block.getFieldValue('red_ports');

  if (red_ports == 'red1'){
    var pin1 = '7';
    var pin2 = '8';
    var pin3 = '12';
  }
  else{
    var pin1 = '0';
    var pin2 = '2';
    var pin3 = '4';
  }

  var stateOutput1 = Blockly.Arduino.valueToCode(block, 'STATE1', Blockly.Arduino.ORDER_ATOMIC) || 'LOW';
  var pinSetupCode1 = 'pinMode(' + pin1 + ', OUTPUT);' + '\n';

  var stateOutput2 = Blockly.Arduino.valueToCode(block, 'STATE2', Blockly.Arduino.ORDER_ATOMIC) || 'LOW';
  var pinSetupCode2 = 'pinMode(' + pin2 + ', OUTPUT);' + '\n';

  var stateOutput3 = Blockly.Arduino.valueToCode(block, 'STATE3', Blockly.Arduino.ORDER_ATOMIC) || 'LOW';
  var pinSetupCode3 = 'pinMode(' + pin3 + ', OUTPUT);' + '\n';

  Blockly.Arduino.addSetup('my_block_' + pin1 + pin2 + pin3, pinSetupCode1 + pinSetupCode2 + pinSetupCode3, false);
  var code = 'digitalWrite(' + pin1 + ', ' + stateOutput1 + '); \n'
    + 'digitalWrite(' + pin2 + ', ' + stateOutput2 + '); \n'
    + 'digitalWrite(' + pin3 + ', ' + stateOutput3 + '); \n';

  return code;
};
```

Εικόνα 5-24. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ LED.

```
Blockly.Arduino['my_block_highlow'] = function(block) {
  var code = block.getFieldValue('STATE');
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-25. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Ανοιχτό/Κλειστό.

```
void setup() {
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(12, LOW);
}
```

Εικόνα 5-26. Παραγόμενος κώδικας C++ μπλοκ LED.

## 5.2.6 Μπλοκ Buttons

Στην Εικόνα 5.27 παρουσιάζεται ο κώδικας για τη δημιουργία του οπτικού μπλοκ των buttons αλλά και τα τρία μπλοκ των κουμπιών ξεχωριστά, ενώ στην Εικόνα 5.28 το αποτέλεσμα αυτού.

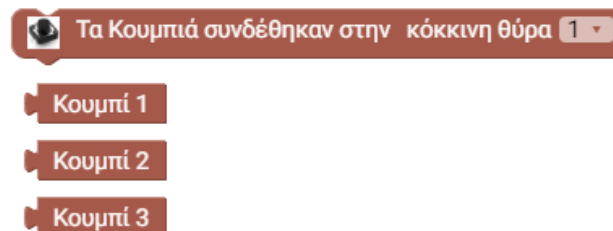
```
Blockly.Blocks['my_block_buttons'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(10);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/button.png", 20, 20, ""))
      .appendField("Τα Κουμπιά συνδέθηκαν στην")
      .appendField(new Blockly.FieldDropdown([["κόκκινη θύρα 1", "red1"], ["κόκκινη θύρα 2", "red2"]], 'red_ports'));
    this.setTooltip("Όρισε σε ποιά κόκκινη θύρα σύνδεσες τα τρία κουμπιά.");
  },
};

Blockly.Blocks['my_block_button1'] = {
  init: function() {
    this.setColour(10);
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.appendDummyInput()
      .appendField("Κουμπι 1");
    this.setTooltip("Όταν πατάς το κουμπι 1 δίνει απάντηση ΝΑΙ, όταν δεν το πατάς δίνει απάντηση ΟΧΙ.");
  },
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};

Blockly.Blocks['my_block_button2'] = {
  init: function() {
    this.setColour(10);
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.appendDummyInput()
      .appendField("Κουμπι 2");
    this.setTooltip("Όταν πατάς το κουμπι 2 δίνει απάντηση ΝΑΙ, όταν δεν το πατάς δίνει απάντηση ΟΧΙ.");
  },
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};

Blockly.Blocks['my_block_button3'] = {
  init: function() {
    this.setColour(10);
    this.setOutput(true, Blockly.Types.BOOLEAN.output);
    this.appendDummyInput()
      .appendField("Κουμπι 3");
    this.setTooltip("Όταν πατάς το κουμπι 3 δίνει απάντηση ΝΑΙ, όταν δεν το πατάς δίνει απάντηση ΟΧΙ.");
  },
  getBlockType: function() {
    return Blockly.Types.BOOLEAN;
  }
};
```

Εικόνα 5-27. Κώδικας JavaScript για την δημιουργία του μπλοκ Buttons.



Εικόνα 5-28. Εικονικό μπλοκ Buttons.

Για την χρήση των διακοπών, στο λογισμικό είναι απαραίτητη η χρήση και των τεσσάρων εικονικών μπλοκ της Εικόνας 5.16, αν χρησιμοποιούνται και τα τρία κουμπιά που διατίθενται. Ο κώδικας, όπως φαίνεται στην Εικόνα 5.29, διαβάζει την κατάσταση τριών κουμπιών που είναι συνδεδεμένα σε τρεις ακροδέκτες. Στη συνάρτηση `setup()`, οι ακροδέκτες αυτοί ορίζονται ως είσοδοι με ενεργοποιημένες τις εσωτερικές αντιστάσεις pull-up. γεγονός που σημαίνει ότι τα κουμπιά θεωρούνται κανονικά ανοιχτά (HIGH) και όταν πατηθούν, θα γίνουν κλειστά (LOW). Στη συνάρτηση `loop()`, οι καταστάσεις των κουμπιών διαβάζονται μέσω της `digitalRead()` και αναστρέφονται με την εντολή `!`, ώστε οι μεταβλητές `button1`, `button2` και `button3` να έχουν την τιμή 1 όταν πατηθεί το αντίστοιχο κουμπί και 0 όταν είναι

ανενεργό. Δημιουργήθηκαν και τα τρία μπλοκ Κουμπί 1/2/3 ώστε να μπορεί ο χρήστης να τα χρησιμοποιήσει σε συνθήκες απόφασης.

Στην Εικόνα 5.29 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.30 όπου παρουσιάζεται και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_buttons'] = function(block) {
  var red_ports = block.getFieldValue('red_ports');

  if (red_ports == 'red1'){
    var pin1 = '7';
    var pin2 = '8';
    var pin3 = '12';
  }
  else if(red_ports == 'red2'){
    var pin1 = '0';
    var pin2 = '2';
    var pin3 = '4';
  }
  };

  var pinSetupCode = 'pinMode(' + pin1 + ', INPUT_PULLUP);\n'
    + 'pinMode(' + pin2 + ', INPUT_PULLUP);\n'
    + 'pinMode(' + pin3 + ', INPUT_PULLUP);';
  Blockly.Arduino.addSetup('my_block_' + pin1 + pin2 + pin3, pinSetupCode, false);
  Blockly.Arduino.addDeclaration('my_block_', 'int button1 = 0;\n' + 'int button2 = 0;\n' + 'int button3 = 0;\n');

  var code1 = 'int button1 = !digitalRead('+pin1+');\n';
  var code2 = 'int button2 = !digitalRead('+pin2+');\n';
  var code3 = 'int button3 = !digitalRead('+pin3+');\n';

  var code = code1+code2+code3;
  return code;
};

Blockly.Arduino['my_block_button1'] = function(block) {
  var code = 'button1';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};

Blockly.Arduino['my_block_button2'] = function(block) {
  var code = 'button2';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};

Blockly.Arduino['my_block_button3'] = function(block) {
  var code = 'button3';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
```

Εικόνα 5-29. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Buttons.

```
int button1 = 0;
int button2 = 0;
int button3 = 0;

void setup() {
  pinMode(7, INPUT_PULLUP);
  pinMode(8, INPUT_PULLUP);
  pinMode(12, INPUT_PULLUP);
}

void loop() {
  int button1 = !digitalRead(7);
  int button2 = !digitalRead(8);
  int button3 = !digitalRead(12);
}
```

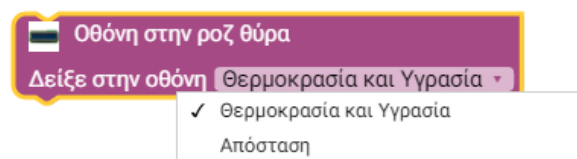
Εικόνα 5-30. Παραγόμενος κώδικας C++ μπλοκ Buttons.

### 5.2.7 Μπλοκ LCD

Στην Εικόνα 5.31 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ των buttons αλλά και τα τρία μπλοκ των κουμπιών, ενώ στην Εικόνα 5.32 το αποτέλεσμα του.

```
Blockly.Blocks['my_block_I2C_lcd'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(322);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/lcd.jpg", 20, 20, ""))
      .appendField("Οθόνη στην ροζ θύρα")
    this.appendDummyInput()
      .appendField("Δείξε στην οθόνη")
      .appendField(new Blockly.FieldDropdown([["Θερμοκρασία και Υγρασία", "temp_hum"], ["Απόσταση", "dis"]], 'pink_ports'));
    this.setTooltip("Επέλεξε τι θέλεις να δείξει η οθόνη.");
  }
};
```

Εικόνα 5-31. Κώδικας JavaScript για την δημιουργία του μπλοκ LCD.



Εικόνα 5-32. Εικονικό μπλοκ LCD.

Για την χρήση της οθόνης στο λογισμικό ώστε να εμφανίζει δεδομένα, είναι απαραίτητη η χρήση του μπλοκ της Εικόνας 5.32. Όπως φαίνεται και στην Εικόνα 5.34, με τη χρήση του μπλοκ αρχικά, γίνεται εισαγωγή της βιβλιοθήκης LiquidCrystal\_I2C.h όπου παρέχει τις απαραίτητες λειτουργίες για τον χειρισμό της οθόνης LCD. Η σύνδεση γίνεται μέσω του πρωτοκόλλου I2C, με διεύθυνση 0x27. Στη συνάρτηση setup(), η οθόνη αρχικοποιείται και ενεργοποιείται ο οπίσθιος φωτισμός. Στη συνάρτηση loop(), ο κώδικας τοποθετεί τον κέρσορα

στις κατάλληλες θέσεις και εμφανίζει στη γραμμή 0 την ένδειξη “Θερμοκρασία” και την τιμή temp και στη γραμμή 1 την ένδειξη “Υγρασία” και την τιμή hum όπου θα του έχει δώσει ο αισθητήρας θερμοκρασίας και υγρασίας. Αν ο χρήστης επιλέξει την ‘Απόσταση’ θα του εμφανίσει το αντίστοιχο μήνυμα και την τιμή που θα έχει διαβαστεί από τον αισθητήρα απόστασης.

Στην Εικόνα 5.33 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.34 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_I2C_lcd'] = function(block) {
  var pink_ports = block.getFieldValue('pink_ports');

  if (pink_ports == 'temp_hum'){ //temperature and humidity
    var code1 = 'lcd.setCursor(0,0);\n';
    var code2 = 'lcd.print("Thermokrasia:");\n';
    var code3 = 'lcd.setCursor(0,1);\n';
    var code4 = 'lcd.print("Ygrasia:");\n';
  }
  else{ //distance
    var code1 = 'lcd.setCursor(0,0);\n';
    var code2 = 'lcd.print("Apostash:");\n';
    var code3 = 'lcd.setCursor(0,1);\n';
    var code4 = 'lcd.print("cm");\n';
  }

  var lcdName = 'lcd';

  Blockly.Arduino.addInclude('my_block', '//Library for I2C LCD\n'+ '#include <Wire.h>\n'+ '#include <LiquidCrystal_I2C.h>');
  Blockly.Arduino.addDeclaration('my_block_', '//set the LCD address to 0x27 for a 16 chars and 2 line display\n'+
    '+LiquidCrystal_I2C '+ lcdName + '(0x27,20,4)' + ';\n');

  var pinSetupCode = lcdName + '.init(); //initialize the lcd\n'+ 'lcd.init();\n'+ 'lcd.backlight();\n';

  Blockly.Arduino.addSetup(lcdName, pinSetupCode, true);

  var code=code1+code2+code3+code4;
  return code;
};
```

Εικόνα 5-33. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ LCD.

```
//Library for I2C LCD
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//set the LCD address to 0x27 for a 16 chars
LiquidCrystal_I2C lcd(0x27,20,4);

void setup() {
  lcd.init(); //initialize the lcd
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.setCursor(0,0);
  lcd.print("Thermokrasia:");
  lcd.print(temp);
  lcd.setCursor(0,1);
  lcd.print("Ygrasia:");
  lcd.print(hum);
}
```

Εικόνα 5-34. Παραγόμενος κώδικας C++ μπλοκ LCD.

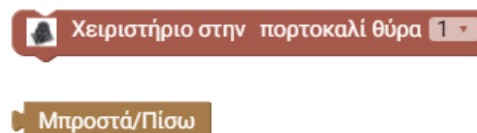
### 5.2.8 Μπλοκ Joystick

Στην Εικόνα 5.35 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ των buttons, αλλά και τα τρία μπλοκ των κουμπιών ξεχωριστά, ενώ στην Εικόνα 5.36 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_joystick'] = {
  init: function() {
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(10);
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/joystick.png", 20, 20, ""))
      .appendField("Χειριστήριο στην")
      .appendField(new Blockly.FieldDropdown([["πορτοκαλί θύρα 1", "or1"], ["πορτοκαλί θύρα 2", "or2"]]), 'orange_ports');
    this.setTooltip("Όρισε σε ποιά πορτοκαλί θύρα σύνδεσες το χειριστήριο.");
  }
};

Blockly.Blocks['my_block_frontback_variable'] = {
  init: function() {
    this.setColour(32);
    this.setOutput(true, Blockly.Types.NUMBER.output);
    this.appendDummyInput()
      .appendField("Μπροστά/Πίσω")
    this.setTooltip("Αποτέλεσμα χειριστήριου αναλογα αν το γυρνάς μπροστά η πίσω.");
  },
  getBlockType: function() {
    return Blockly.Types.NUMBER;
  }
};
```

Εικόνα 5-35. Κώδικας JavaScript για την δημιουργία του μπλοκ Joystick.



Εικόνα 5-36. Εικονικό μπλοκ Joystick.

Για την χρήση του μοχλού στο λογισμικό είναι απαραίτητη η χρήση του μπλοκ της Εικόνας 5.36. Όπως φαίνεται και στην Εικόνα 5.38, με τη χρήση του μπλοκ ο κώδικας διαβάζει τις αναλογικές τιμές από τον μοχλό. Οι δυο ακροδέκτες αντιστοιχούν στους άξονες X και Y. Στη συνάρτηση loop(), ο κώδικας διαβάζει τις αναλογικές τιμές του μοχλού και τις αποθηκεύει σε δυο μεταβλητές. Οι τιμές αυτές κυμαίνονται συνήθως από 0 έως 1023 και αντιπροσωπεύουν τη θέση του σε κάθε άξονα, με τη μέση τιμή (περίπου 512) να δείχνει ότι ο μοχλός είναι στη θέση ηρεμίας. Δημιουργήθηκε και το μπλοκ Μπροστά/Πίσω για χρήση με κάποιο κινητήρα.

Στην Εικόνα 5.37 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.38 όπου παρουσιάζεται και ο κώδικας που βλέπει ο χρήστης.



```
Blockly.Arduino['my_block_joystick'] = function(block) {
  var orange_ports = block.getFieldValue('orange_ports');

  if (orange_ports == 'or1'){
    var VRX_PIN = 'A0';
    var VRY_PIN = 'A1';
  }
  else if(orange_ports == 'or2'){
    var VRX_PIN = 'A6';
    var VRY_PIN = 'A7';
  }
  };

  Blockly.Arduino.addDeclaration(VRX_PIN + VRY_PIN, 'const int VRX_PIN = ' + VRX_PIN + ';      // joystick X axis\n'
    + 'const int VRY_PIN = ' + VRY_PIN + ';      // joystick Y axis\n'
    + 'int xValue = 0; // To store value of the X axis\n'
    + 'int yValue = 0; // To store value of the Y axis\n'
    );

  var code='// read analog X and Y analog values\n xValue = analogRead(VRX_PIN);\n yValue = analogRead(VRY_PIN);\n';

  return code;
};

Blockly.Arduino['my_block_frontback_variable'] = function(block) {
  var code = 'xValue';
  return [code, Blockly.Arduino.ORDER_ATOMIC];
};
};
```

Εικόνα 5-37. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ Joystick.

```
const int VRX_PIN = A0;      // joystick X axis
const int VRY_PIN = A1;      // joystick Y axis
int xValue = 0; // To store value of the X axis
int yValue = 0; // To store value of the Y axis

void setup() {
}

void loop() {
  // read analog X and Y analog values
  xValue = analogRead(VRX_PIN);
  yValue = analogRead(VRY_PIN);
}
```

Εικόνα 5-38. Παραγόμενος κώδικας C++ μπλοκ Joystick.

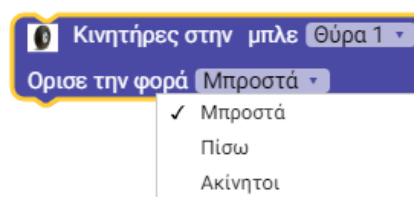
## Επενεργητές

### 5.2.9 Μπλοκ DC Motor

Στην Εικόνα 5.39 παρουσιάζεται ο κώδικας για την δημιουργία του οπτικού μπλοκ των buttons αλλά και τα τρία μπλοκ των κουμπιών ξεχωριστά, ενώ στην Εικόνα 5.40 το αποτέλεσμα αυτού.

```
Blockly.Blocks['my_block_motor'] = {
  init: function() {
    this.appendDummyInput()
      .appendField(new Blockly.FieldImage("img/ColourBot/wheel.jpg", 20, 20, ""))
      .appendField("Κινητήρες στην")
      .appendField(new Blockly.FieldDropdown([["μπλε θύρα 1", "blue1"], ["μπλε θύρα 2", "blue2"]]), 'blue_ports');
    this.appendDummyInput()
      .appendField("Όρισε την φορά")
      .appendField(new Blockly.FieldDropdown([["Μπροστά", "front"], ["Πίσω", "back"], ["Ακίνητοι", "stop"]]), "rotation");
    /*this.appendValueInput("pwm_speed")
      .setCheck(Blockly.Types.NUMBER.checkList)
      .appendField("και την ταχύτητα με τιμή (0-255)");*/
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setColour(240);
    this.setTooltip("Όρισε αν ο κινητήρας θα πηγαίνει μπροστά ή πίσω και με πόση ταχύτητα");
  }
};
```

Εικόνα 5-39. Κώδικας JavaScript για την δημιουργία του μπλοκ DC Motor.



Εικόνα 5-40. Εικονικό μπλοκ DC Motor.

Για την χρήση του κινητήρα στο λογισμικό είναι απαραίτητη η χρήση του εικονικού μπλοκ της Εικόνας 5.40. Με τη χρήση του μπλοκ χρησιμοποιούνται δύο ακροδέκτες του Arduino, έναν για αναλογική έξοδο PWM και έναν για ψηφιακή έξοδο, όπως φαίνεται και στην Εικόνα 5.42. Στην συνάρτηση `setup()`, οι ακροδέκτες ορίζονται ως έξοδοι (OUTPUT). Στη συνέχεια, στη συνάρτηση `loop()`, θέτει έναν ακροδέκτη σε κατάσταση LOW και εφαρμόζει μέγιστη τάση (255, που αντιστοιχεί στο 100% του PWM) στον άλλο ακροδέκτη. Σύμφωνα με τον πίνακα που έχει παρουσιαστεί ανάλογα αν επιλεγεί Μπροστά/Πίσω/Ακίνητοι αλλάζουν και οι τιμές που παίρνουν οι ακροδέκτες στον κώδικα.

Στην Εικόνα 5.41 παρουσιάζεται ο κώδικας για την δημιουργία του παραγόμενου κώδικα που θα χρησιμοποιηθεί από το Arduino IDE όπως αναλύθηκε παραπάνω όταν χρησιμοποιηθεί το μπλοκ. Το αποτέλεσμα του φαίνεται στην Εικόνα 5.42 όπου είναι και ο κώδικας που βλέπει ο χρήστης.

```
Blockly.Arduino['my_block_motor'] = function(block) {
  var blue_ports = block.getFieldValue('blue_ports');
  var rotation = block.getFieldValue('rotation');
  //var speed = Blockly.Arduino.valueToCode(block, 'pwm_speed', Blockly.Arduino.ORDER_ATOMIC || '0');

  if (blue_ports == 'blue1'){
    var pin1 = '13';
    var pin2 = '3';
  }
  else if(blue_ports == 'blue2'){
    var pin1 = '5';
    var pin2 = '6';
  };

  Blockly.Arduino.setups_['pin1'] = 'pinMode(' + pin1 + ', OUTPUT);';
  Blockly.Arduino.setups_['pin2'] = 'pinMode(' + pin2 + ', OUTPUT);';

  if (rotation == 'front')
  {
    var code1 = 'digitalWrite(' + pin1 + ', ' + '0' + ');\\n';
    var code2 = 'analogWrite(' + pin2 + ', ' + '255' + ');\\n';
  }
  else if(rotation == 'stop'){
    var code1 = 'digitalWrite(' + pin1 + ', ' + '0' + ');\\n';
    var code2 = 'analogWrite(' + pin2 + ', ' + '0' + ');\\n';
  }
  else
  {
    var code1 = 'digitalWrite(' + pin1 + ', ' + '255' + ');\\n';
    var code2 = 'analogWrite(' + pin2 + ', ' + '0' + ');\\n';
  };

  var code =code1+code2;
  return code;
};
```

Εικόνα 5-41. Κώδικας JavaScript για τον κώδικα C++ που παράγεται όταν χρησιμοποιείται το μπλοκ DC Motor.

```
void setup() {
  pinMode(3, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, 0);
  analogWrite(3, 255);
}
```

Εικόνα 5-42. Παραγόμενος κώδικας C++ μπλοκ DC Motor.

## ΚΕΦΑΛΑΙΟ 6: ΤΡΙΣΔΙΑΣΤΑΤΑ ΚΕΛΥΦΗ

---

Δεδομένου ότι ο πρωταρχικός στόχος του συστήματος είναι να παρέχει μια πλατφόρμα που επιτρέπει στους χρήστες να δημιουργούν λειτουργικά αντικείμενα, είναι σημαντικό να διασφαλιστεί η ύπαρξη κατάλληλων κελυφών για τα ηλεκτρονικά εξαρτήματα καθώς και ενός αξιόπιστου μηχανισμού σύνδεσης<sup>3</sup>. Στο σύστημα που περιγράφεται στην εργασία [5], υιοθετήθηκε η χρήση λειτουργικών κελυφών τυπωμένων σε 3D για τα ηλεκτρονικά εξαρτήματα. Ωστόσο, ένα βασικό πρόβλημα που εντοπίστηκε ήταν ότι τα στοιχεία σύνδεσης ήταν ευαίσθητα σε φθορά μετά από εκτεταμένη χρήση, αποτελώντας έτσι εμπόδιο στη μακροχρόνια ανθεκτικότητα της πλατφόρμας. Επιπλέον, τα ενσωματωμένα σημεία αγκύρωσης προσέφεραν μόνο μία επιλογή για τη σύνδεση των συσκευών, περιορίζοντας την ευελιξία των συνδέσεων.

Για την αντιμετώπιση των παραπάνω ζητημάτων, στο προτεινόμενο σύστημα υιοθετήθηκε μια παρόμοια προσέγγιση με τη χρήση εξαρτημάτων τυπωμένων σε 3D, αλλά με διαφορετικά βασικά χαρακτηριστικά.

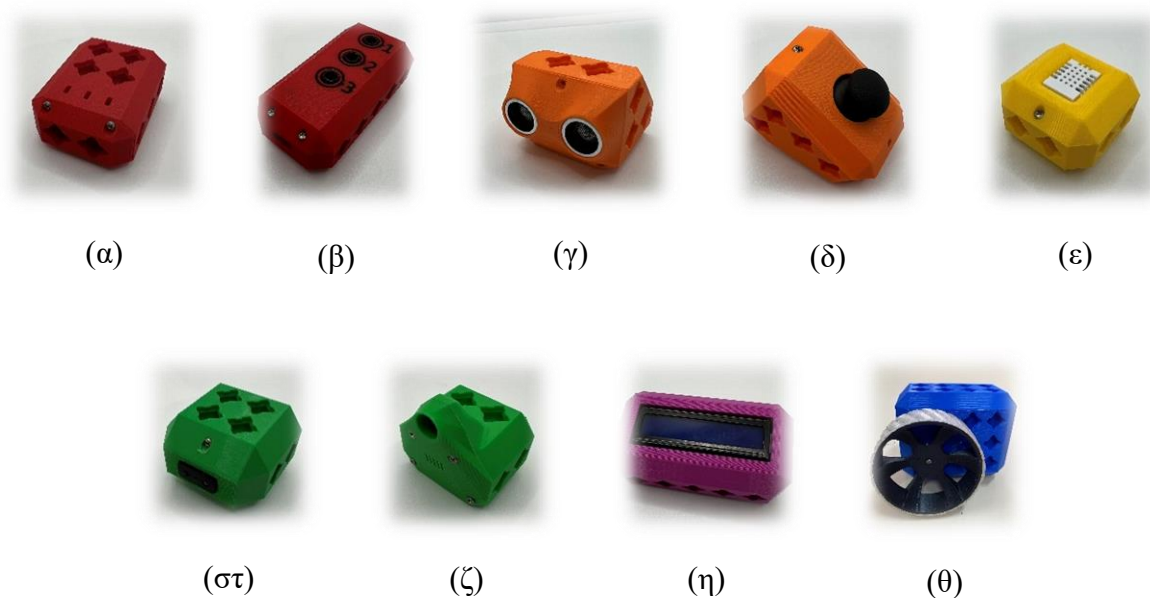
### 6.1 Τρισδιάστατα κελύφη συστήματος

Κλειδί του προτεινόμενου συστήματος είναι ο τρόπος που περιβάλλονται τα ηλεκτρονικά στοιχεία. Έτσι, προτείνεται η έννοια των λειτουργικών κελυφών. Σκοπός είναι να υπάρχει η δυνατότητα να συνδέονται διάφοροι τύποι ηλεκτρονικών συσκευών προκειμένου να δημιουργηθεί οποιαδήποτε συσκευή μπορεί να φανταστεί ο χρήστης. Για να επιτευχθεί αυτό, κατασκευάστηκαν τα κελύφη με πρωτότυπη προσέγγιση, χρησιμοποιώντας συγκεκριμένους περιορισμούς για κάθε ηλεκτρονικό στοιχείο. Αποφεύγονται σοβαρά προβλήματα, όπως περιορισμένες επιλογές σύνδεσης, που μπορεί να προκαλέσουν σημαντικά εμπόδια στην ευελιξία της πλατφόρμας. Χρησιμοποιήθηκε υψηλής ποιότητας 3D εκτυπωτική συσκευή με υλικό PLA.

---

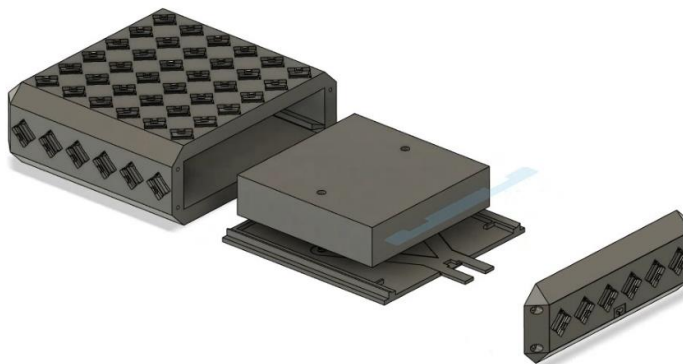
<sup>3</sup> Τα τρισδιάστατα κελύφη σχεδιάστηκαν και κατασκευάστηκαν από τον κ. Γ. Κυπραίο, αλλά παρουσιάζονται συνοπτικά σε αυτή την εργασία για λόγους πληρότητας, προκειμένου να γίνει κατανοητή η μεθοδολογία σύνδεσης των δομικών στοιχείων και η δημιουργία λειτουργικών συσκευών.

Στην Εικόνα 6.1 παρουσιάζεται ο σχεδιασμός του κελύφους των αρθρωμάτων του συστήματος. Ο προτεινόμενος σχεδιασμός επικεντρώνεται στην προσαρμοστικότητα, την προστασία και την ανθεκτικότητα, διασφαλίζοντας ότι οι χρήστες μπορούν να ασχολούνται με ηλεκτρονικά κυκλώματα και να δημιουργούν λειτουργικά πρωτότυπα με σιγουριά, γνωρίζοντας ότι η πλατφόρμα είναι ανθεκτική και αξιόπιστη.

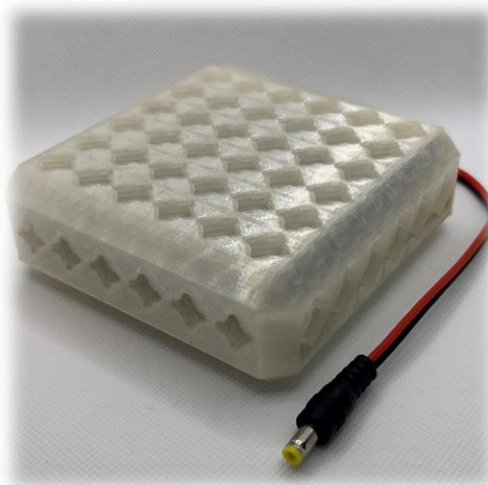


Εικόνα 6-1. Κελύφη (α) LED, (β) Buttons, (γ) Sonar, (δ) Joystick, (ε) Temperature & Humidity, (στ) Infrared, (ζ) Line Tracking, (η) LCD και (θ) DC Motor.

Στη Εικόνα 6.2 παρουσιάζεται η συναρμολόγηση της μονάδας τροφοδοσίας με το λειτουργικό κέλυφος που χρησιμοποιείται για την περίκλεισή της και στην Εικόνα 6.3 το εκτυπωμένο 3D πρωτότυπο.



Εικόνα 6-2. Σχέδιο κελύφους Μονάδας Τροφοδοσίας.



Εικόνα 6-3. Εκτυπωμένο 3D πρωτότυπο Μονάδας Τροφοδοσίας.

Επιπλέον, ενσωματώνεται μια επιπλέον στρώση προστασίας με τη μορφή εξωτερικών ελαστικών χειλών για το κύριο κέλυφος. Αυτή η προστατευτική λειτουργία δρα ως μια ισχυρή ασπίδα για την κύρια πλακέτα (Εικόνα 6.4), που φιλοξενείται εντός του κελύφους, μειώνοντας αποτελεσματικά την επίδραση των πτώσεων και ενισχύοντας τη συνολική ανθεκτικότητα του συστήματος. Στην Εικόνα 6.5 παρουσιάζεται το εκτυπωμένο άρθρωμα για την κεντρική πλακέτα.



Εικόνα 6-4. Προστατευτικά τμήματα Κεντρικής Μονάδας.



Εικόνα 6-5. Εκτυπωμένο 3D πρωτότυπο Κεντρικής Μονάδας.

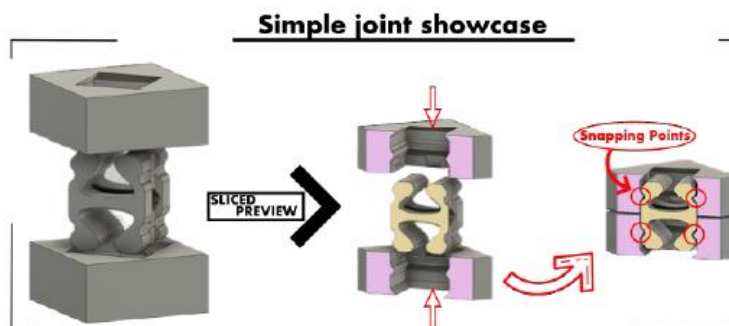
Ο χρωματικός κώδικας που αναφέρθηκε παραπάνω συνδέεται άμεσα με τα κελύφη. Όλα τα αρθρώματα έχουν χρώμα με σκοπό να συνδέονται στις αντίστοιχες αριθμημένες και χρωματιστές θύρες στην κεντρική πλακέτα (Εικόνα 6.6) και με ανάλογο τρόπο στο λογισμικό.



Εικόνα 6-6. Χρωματικός κώδικας και κελύφη.

## 6.2 Μηχανισμός σύνδεσης

Ένας άλλος κρίσιμος παράγοντας είναι το πώς μπορούν να συνδέονται αυτά τα λειτουργικά κελύφη έτσι ώστε να δημιουργηθούν πολύπλοκες συσκευές. Αυτό που προτείνεται είναι ένας νέος μηχανισμός σύνδεσης, ο οποίος διασφαλίζει τη δυνατότητα σύνδεσης σε οποιαδήποτε κατεύθυνση με έναν ανθεκτικό τρόπο, όπως φαίνεται στην Εικόνα 6.4.



Εικόνα 6-4. Σχεδιασμός δομικών στοιχείων.

Αυτή η προσέγγιση χρησιμοποιεί έναν απλό μηχανισμό σύνδεσης (Εικόνα 6.5) με μια πρόσθετη εξωτερική υποδοχή, η οποία επιτρέπει τη σύνδεση διαφορετικών λειτουργικών κελυφών. Η ίδια υποδοχή μπορεί να χρησιμοποιηθεί σε κάθε εξωτερικό σημείο σύνδεσης οποιουδήποτε λειτουργικού κελύφους. Η προτεινόμενη προσέγγιση ενσωματώνει την αρχή σχεδίασης που παρουσιάστηκε αρχικά στο σύστημα PrintABlok [26].



Εικόνα 6-5. Μηχανισμός σύνδεσης.



## ΚΕΦΑΛΑΙΟ 7: ΕΦΑΡΜΟΓΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

---

Αφού ολοκληρώθηκε η κατασκευή των βασικών πυλώνων, ακολούθησε μια μελέτη για τον σχεδιασμό ενδεικτικών διατάξεων που θα μπορούσαν να συναρμολογήσουν οι μαθητές. Ο στόχος αυτής της φάσης ήταν διττός: αφενός, να επιβεβαιωθεί ότι οι κατασκευές μπορούν να επιτύχουν την επιθυμητή ποικιλία και πολυπλοκότητα, προσφέροντας στους μαθητές πλούσιες δυνατότητες δημιουργίας και πειραματισμού· αφετέρου, να διαμορφωθούν συγκεκριμένα πρότζεκτ που θα λειτουργούσαν ως προκλήσεις-στόχοι για τους μαθητές, ενθαρρύνοντάς τους να επιλύσουν προβλήματα και να αναπτύξουν τις δεξιότητές τους. Οι ενδεικτικές εφαρμογές, σχεδιάστηκαν έτσι ώστε να είναι προσαρμοσμένα στα διάφορα επίπεδα δυσκολίας, εμπνέοντας τους μαθητές και καλλιεργώντας την αυτοπεποίθησή τους στην κατασκευαστική διαδικασία.

### **7.1 Ενδεικτικές εφαρμογές**

#### **7.1.1 Ρυθμιζόμενα LED από κουμπιά**

Στο πρώτο παράδειγμα (Εικόνα 7.1), κατασκευάστηκε μια εφαρμογή που αποτελείται από έξι δομικά μέρη, τα οποία είναι τα εξής:

1. Κεντρική Μονάδα
2. Μονάδα Τροφοδοσίας
3. Άρθρωμα LED
4. Άρθρωμα Buttons
5. Σύνδεσμοι
6. Καλώδια σύνδεσης



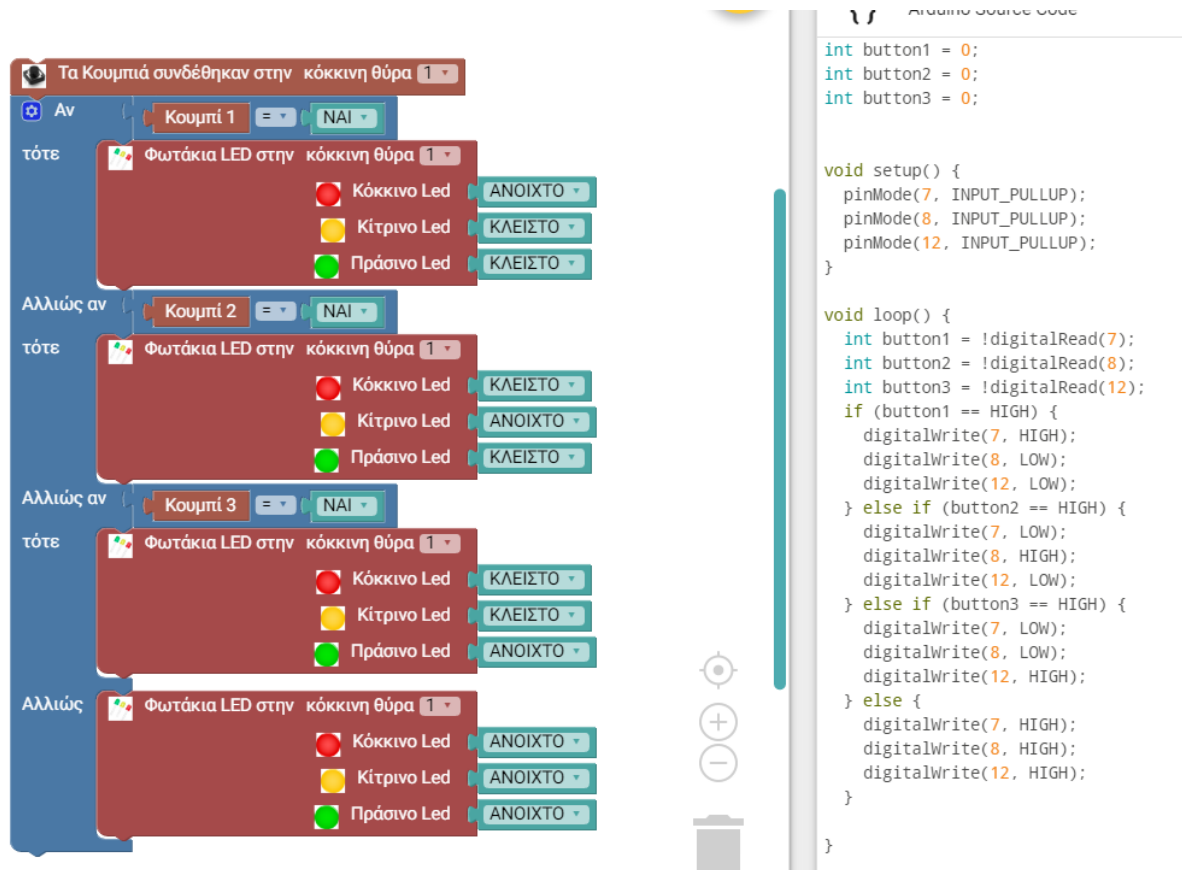
Εικόνα 7-1. Αποσυναρμολογημένη διάταξη ρυθμιζόμενων LED από κουμπιά.

Η ολοκληρωμένη μορφή της συναρμολογημένης διάταξης απεικονίζεται στην Εικόνα 7.2.



Εικόνα 7-2. Απεικόνιση διάταξης ρυθμιζόμενων led από κουμπιά.

Για να λειτουργήσει η διάταξη, αναπτύχθηκε ένας ενδεικτικός κώδικας προγραμματισμού (Εικόνα 7.3). Ο κώδικας αυτός έχει σχεδιαστεί ώστε να επιδεικνύει τη σωστή λειτουργία και την ευχέρεια συνδεσιμότητας, επιτρέποντας την εναλλαγή των συνδυασμών των LED, τα οποία ανάβουν ανάλογα με το κουμπί που πιέζεται.



Εικόνα 7-3. Παράδειγμα κώδικα ρυθμιζόμενων LED από κουμπιά.

### 7.1.2 Εμφάνιση δεδομένων στην οθόνη

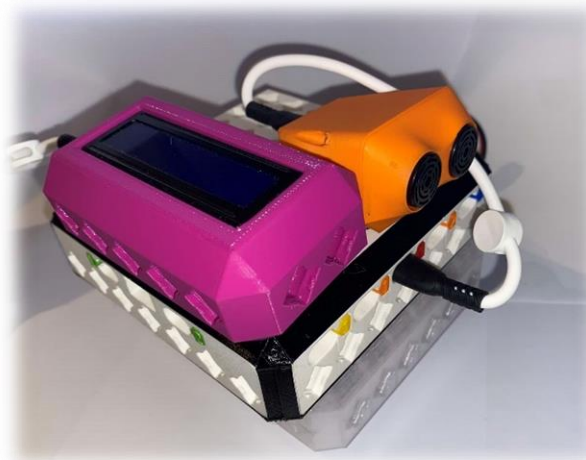
Στο δεύτερο παράδειγμα (Εικόνα 7.4), κατασκευάστηκε μια εφαρμογή που αποτελείται από έξι δομικά μέρη, τα οποία είναι τα εξής:

1. Κεντρική Μονάδα
2. Μονάδα Τροφοδοσίας
3. Άρθρωμα LCD
4. Άρθρωμα Sonar Sensor
5. Σύνδεσμοι
6. Καλώδια σύνδεσης



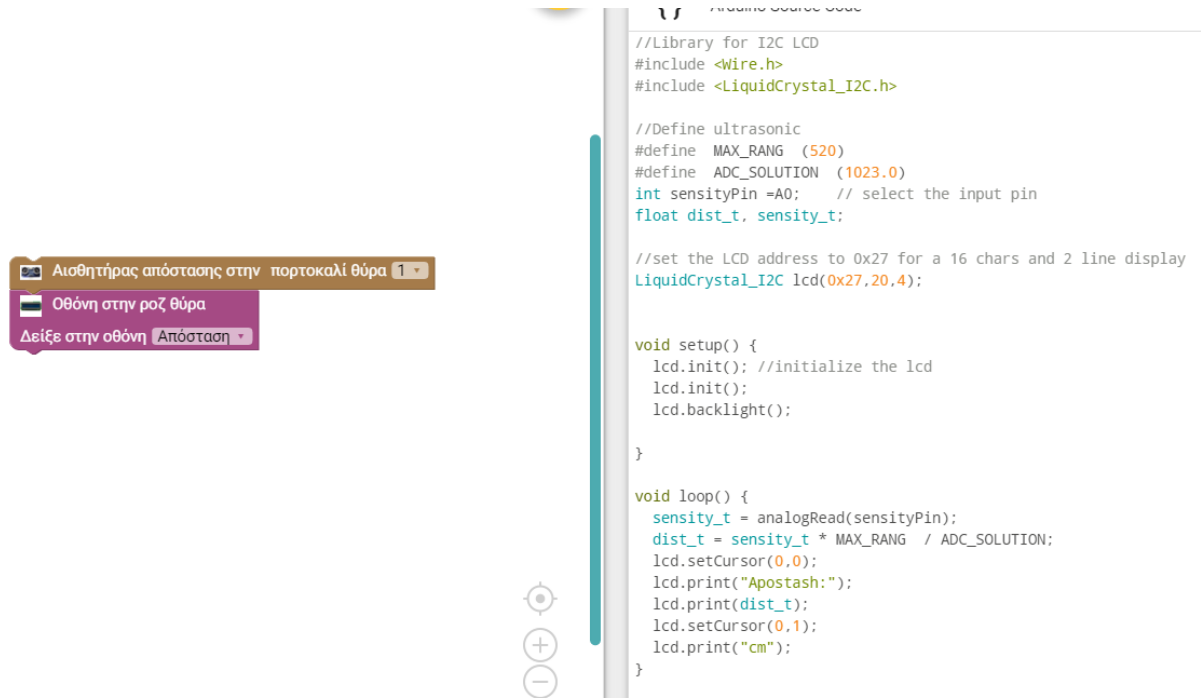
Εικόνα 7-4. Αποσυναρμολογημένη διάταξη για εμφάνιση δεδομένων στην οθόνη.

Η ολοκληρωμένη μορφή της συναρμολογημένης διάταξης απεικονίζεται στην Εικόνα 7.5.



Εικόνα 7-5. Απεικόνιση διάταξης για εμφάνιση δεδομένων στην οθόνη.

Για να λειτουργήσει η διάταξη, αναπτύχθηκε ένας ενδεικτικός κώδικας προγραμματισμού (Εικόνα 7.6). Συγκεκριμένα αναπτύχθηκε ένας κώδικας που επιτρέπει την εμφάνιση της απόστασης που ανιχνεύει ο αισθητήρας από κάποιο εμπόδιο στην οθόνη χαρακτήρων.



Εικόνα 7-6. Παράδειγμα κώδικα για εμφάνιση δεδομένων στην οθόνη.

### 7.1.3 Ανίχνευση εμποδίου

Στο τρίτο παράδειγμα (Εικόνα 7.7), κατασκευάστηκε μια εφαρμογή που αποτελείται από έξι δομικά μέρη, τα οποία είναι τα εξής:

1. Κεντρική Μονάδα
2. Μονάδα Τροφοδοσίας
3. Άρθρωμα LED
4. Άρθρωμα Infrared Sensor
5. Σύνδεσμοι
6. Καλώδια σύνδεσης



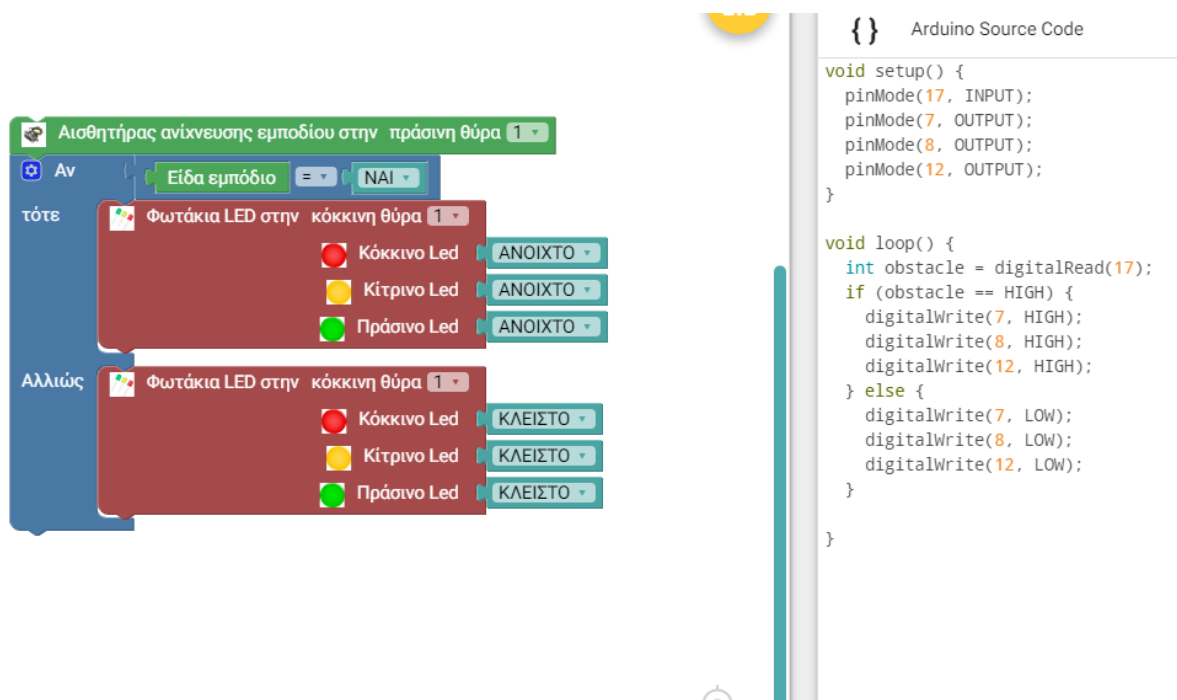
Εικόνα 7-7. Αποσυναρμολογημένη διάταξη για ανίχνευση εμποδίου.

Η ολοκληρωμένη μορφή της συναρμολογημένης διάταξης απεικονίζεται στην Εικόνα 7.8.



Εικόνα 7-8. Απεικόνιση διάταξης για ανίχνευση εμποδίου.

Για να λειτουργήσει η διάταξη, αναπτύχθηκε ένας ενδεικτικός κώδικας προγραμματισμού (Εικόνα 7.9). Συγκεκριμένα αναπτύχθηκε ένας κώδικας που επιτρέπει την ανίχνευση εμποδίου και την ρύθμιση των LED ανάλογα.



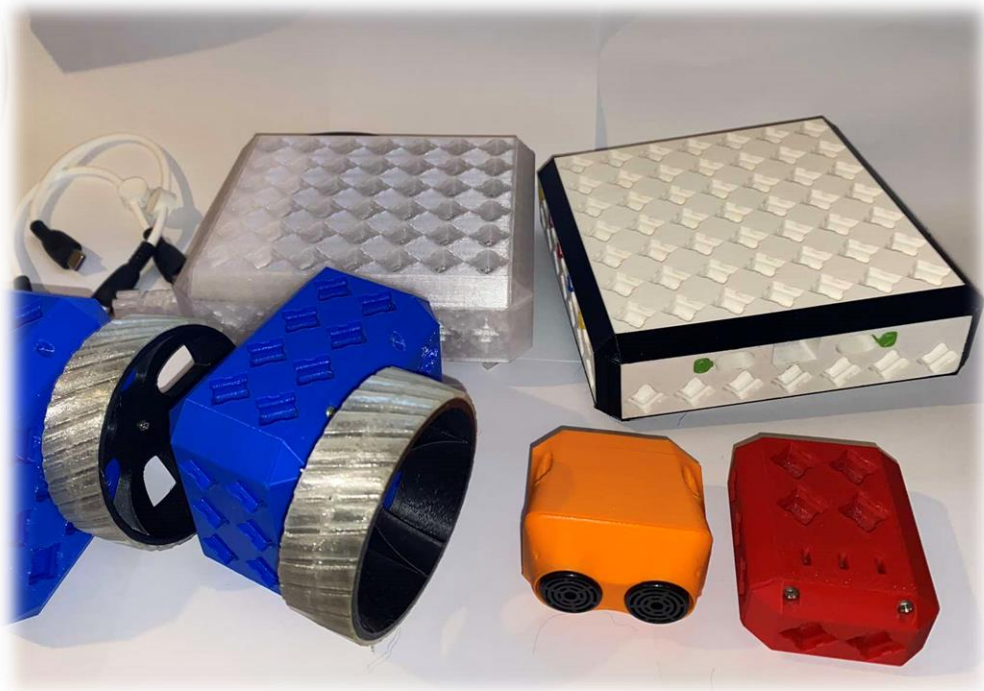
Εικόνα 7-9. Παράδειγμα κώδικα για ανίχνευση εμποδίου.

#### 7.1.4 Ρομποτικό όχημα

Για να αποδειχθεί η εφαρμοσιμότητα της προσέγγισης, δημιουργήθηκε ένα αυτόνομο κινητό ρομπότ. Το ρομπότ αυτό, για να λειτουργήσει, απαιτεί μια σειρά από βασικά εξαρτήματα, όπως κινητήρες που του επιτρέπουν την κίνηση, αισθητήρες που το βοηθούν να αντιλαμβάνεται το περιβάλλον, καθώς και πρόσθετα εξαρτήματα, όπως τα LED για διαδραστικότητα με τους χρήστες. Μια ενδεικτική σύνδεση αυτών των εξαρτημάτων, τα οποία χρησιμοποιήθηκαν για την ανάπτυξη του πρωτοτύπου, παρουσιάζεται στην Εικόνα 7.10. Για την αυτόνομη κίνηση, η μονάδα τροφοδοσίας παρέχει ενέργεια στους αισθητήρες, στους κινητήρες, και στην κεντρική πλακέτα. Κατασκευάστηκε μια εφαρμογή που αποτελείται από επτά δομικά μέρη, τα οποία είναι τα εξής:

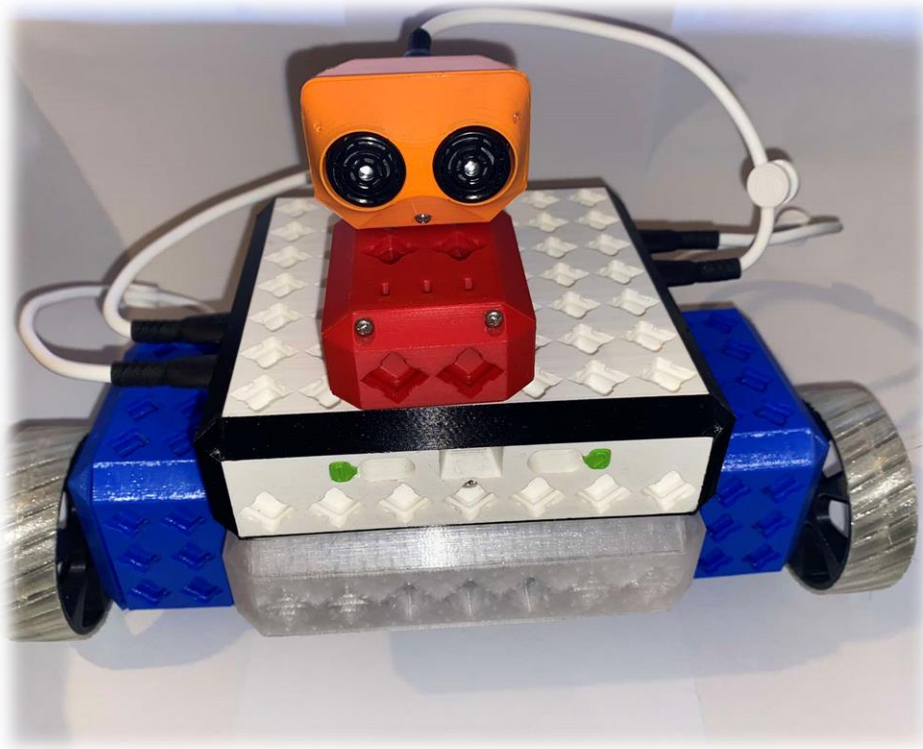
1. Κεντρική Μονάδα
2. Μονάδα Τροφοδοσίας
3. Άρθρωμα LED
4. Άρθρωμα Sonar Sensor
5. Άρθρωμα DC Motor
6. Σύνδεσμοι
7. Καλώδια σύνδεσης





Εικόνα 7-10. Αποσυναρμολογημένη διάταξη κινητού ρομπότ.

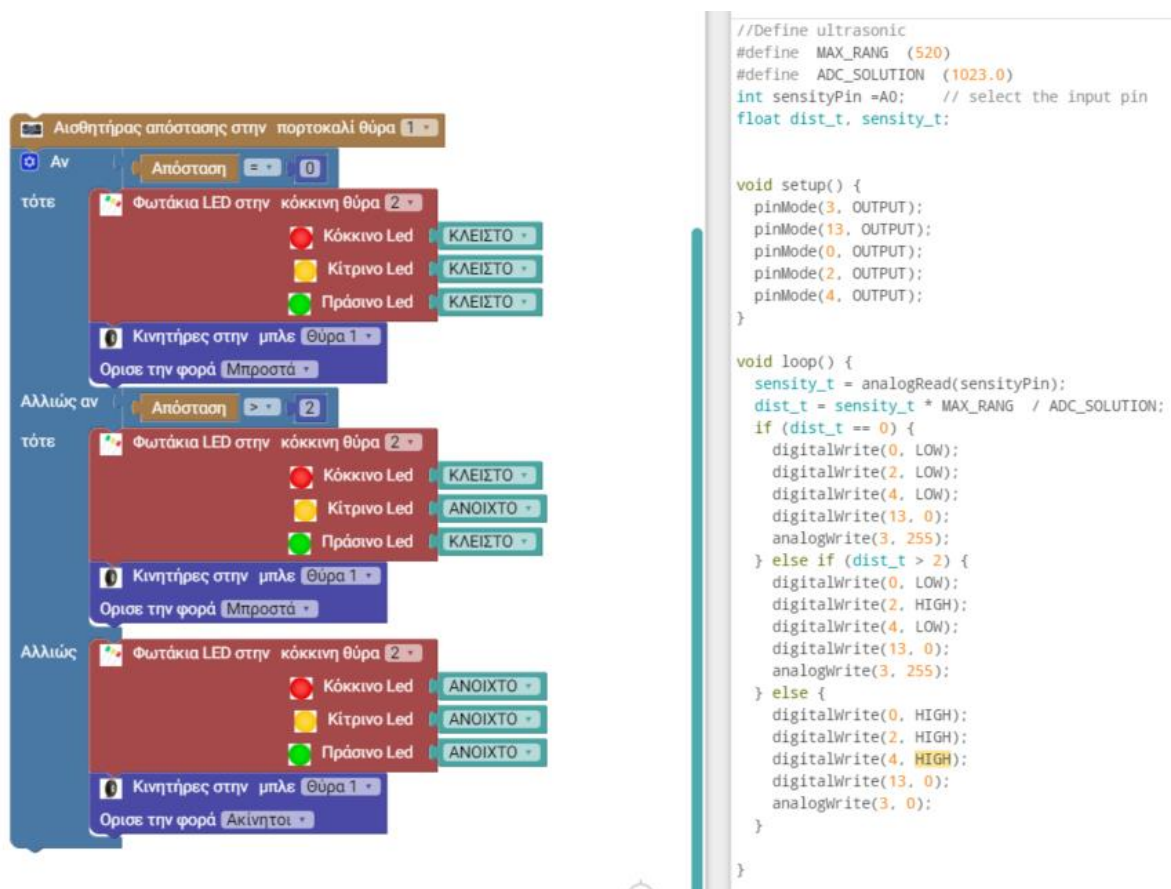
Η ολοκληρωμένη μορφή της συναρμολογημένης διάταξης απεικονίζεται στην Εικόνα 7.11.



Εικόνα 7-11. Απεικόνιση διάταξης για κινητό ρομπότ.

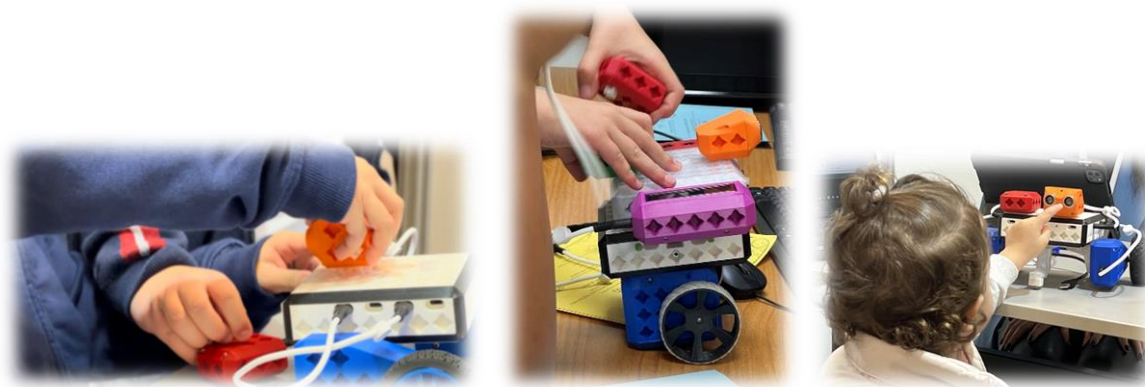


Για να επαληθευτεί η λειτουργικότητα της προσέγγισης, αναπτύχθηκε ένα απλό πρόγραμμα που επιτρέπει στο ρομπότ να ανιχνεύει εμπόδια και να αλληλοεπιδρά με αυτά. Η ανίχνευση εμποδίων πραγματοποιείται μέσω του αισθητήρα σόναρ. Όταν το ρομπότ εντοπίσει εμπόδιο, δηλαδή σταματάει να ανιχνεύει απόσταση ίση με το μηδέν, συνεχίζει με την ίδια ταχύτητα και σταματά όταν η απόσταση είναι μικρότερη από 2 εκατοστά. Επιπλέον, το ρομπότ αλληλοεπιδρά με τον χρήστη μέσω των LED, των οποίων η φωτεινότητα ρυθμίζεται δυναμικά, ανάλογα με την απόσταση του ανιχνευμένου εμποδίου. Στην Εικόνα 7.12 απεικονίζεται η διάταξη του προγράμματος στο Ardublockly μαζί με τον παραγόμενο κώδικα.



Εικόνα 7-12. Παράδειγμα κώδικα για κινητό ρομπότ.

Το αρχικό αυτόνομο ρομπότ δοκιμάστηκε για την ανθεκτικότητα των εξαρτημάτων του μέσω συνεχούς χρήσης, καθώς παρουσιάστηκε σε μια εκδήλωση STEM για τρεις συνεχόμενες ημέρες, όπου παιδιά αλληλοεπιδρούσαν με αυτό. Όπως φαίνεται στην Εικόνα 7.13, τα πρώτα σχόλια ήταν ιδιαίτερα θετικά, προσφέροντας ενθαρρυντικά στοιχεία για την προτεινόμενη μέθοδο.



Εικόνα 7-13. Αλληλεπίδραση παιδιών με το σύστημα.

## ***7.2 Μαθησιακά σενάρια***

Για μια αρχική αξιολόγηση της χρηστικότητας του προτεινόμενου συστήματος σε πραγματικές συνθήκες, επιχειρήθηκε η δοκιμή του από μαθητές της Πρωτοβάθμιας Εκπαίδευσης. Ωστόσο, λόγω περιορισμένου χρόνου για την ολοκλήρωση της εργασίας καθώς και των υποχρεώσεων των μαθητών της Δευτεροβάθμιας Εκπαίδευσης (εξετάσεις και σχολικές εκδρομές), δεν κατέστη εφικτό να διεξαχθεί εκτενής μελέτη με μαθητές αυτής της ηλικιακής ομάδας. Αντ' αυτού, πραγματοποιήθηκαν επιδείξεις σε μικρές ομάδες μαθητών και συζητήσεις με τους εκπαιδευτικούς. Το φύλλο έργου που χρησιμοποιήθηκε στην επίδειξη παρατίθεται αναλυτικά στο Παράρτημα 1.

## ΚΕΦΑΛΑΙΟ 8: ΣΥΜΠΕΡΑΣΜΑΤΑ

---

### 8.1 Συμπεράσματα

Στην παρούσα μελέτη πραγματοποιήθηκε θεωρητική ανάλυση της μεθοδολογίας STEM και ανασκόπηση των βασικών εργαλείων εκπαιδευτικής ρομποτικής. Η μεθοδολογία αυτή, σε συνδυασμό με τα κατάλληλα εργαλεία, κερδίζει συνεχώς έδαφος και ενσωματώνεται σταδιακά στα σύγχρονα εκπαιδευτικά προγράμματα της Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης, καθώς και σε εξωσχολικές δραστηριότητες μάθησης.

Η επισκόπηση αυτή υλοποιήθηκε με σκοπό την ανάδειξη των πλεονεκτημάτων και των εντοπισμό των σημαντικότερων μειονεκτημάτων των συγκεκριμένων εργαλείων, έτσι ώστε η εκπαιδευτική πλατφόρμα που θα σχεδιαζόταν, να επιλύει αυτά τα προβλήματα. Από την μελέτη προέκυψε ότι υπάρχει ανάγκη για σχεδιασμό εκπαιδευτικών εργαλείων, φιλικά ως προς τη χρήση με μειωμένο κόστος, γιατί η απόκτηση γνωστών εργαλείων του εμπορίου καθίσταται συνήθως απαγορευτική (λόγω πολλαπλάσιου κόστους) για τα περισσότερα δημόσια σχολεία, ιδιαίτερα τα τελευταία χρόνια. Παράλληλα, αναγνωρίστηκε η ανάγκη για ένα σχεδιασμό που θα αποτρέπει την πλειοψηφία των λαθών που οφείλονται σε λανθασμένη συνδεσμολογία, καθώς και η αναγκαιότητα για ευκολότερο τρόπο ανάπτυξης του λογισμικού ελέγχου.

Το προτεινόμενο σύστημα μπορεί να χρησιμοποιηθεί ως μέρος ενός πλαισίου για τη διδασκαλία STEM. Ανάλογα πλαίσια, καθώς και οι περιορισμοί και οι αδυναμίες τους, έχουν ήδη συζητηθεί στη βιβλιογραφία [27, 28]. Η προσέγγιση που παρουσιάζεται εδώ έχει σκοπό να είναι ευρύτερη, περιλαμβάνοντας πτυχές από τον σχεδιασμό και την ανάπτυξη των επιμέρους στοιχείων έως τη χρήση και την αξιολόγησή τους σε εκπαιδευτικές δραστηριότητες. Προς τούτο, τα αναπτυγμένα στοιχεία θα αξιολογούνται, θα ενημερώνονται και θα προσαρμόζονται βάσει των σχολίων και των αναγκών των χρηστών (εκπαιδευτικών και μαθητών).

Επιπλέον, με την ενσωμάτωση κατάλληλων μηχανισμών στις διαδικασίες των επιμέρους στοιχείων και με την αξιοποίηση μιας ποικιλίας ψηφιακών εργαλείων (για την προετοιμασία εκπαιδευτικού και μαθησιακού υλικού, δημιουργία δοκιμασιών, ηχογράφηση, βίντεο, απεικόνιση, δημιουργία γραφικών, οργάνωση συνεργατικής διαδικτυακής εργασίας ή διαδικτυακές αναζητήσεις κ.λπ.), διασφαλίζεται ότι το προτεινόμενο πλαίσιο δεν θα είναι

στατικό και παρωχημένο, αλλά θα μπορεί να παραμένει αποτελεσματικό για την επίτευξη διαφορετικών εκπαιδευτικών στόχων υπό διάφορες συνθήκες.

## **8.2 Μελλοντικές Επεκτάσεις**

Η εκπαιδευτική πλατφόρμα με την υφιστάμενη μορφή της, αποτιμήθηκε θετικά από εκπαιδευτικούς και μαθητές και υπάρχουν σκέψεις για περαιτέρω μελλοντική ανάπτυξη. Για να καταστεί δυνατή η ευρεία υιοθέτηση αυτής της προσέγγισης στην εκπαιδευτική διαδικασία, απαιτείται η εκτέλεση επιπλέον βημάτων. Αυτά περιλαμβάνουν τόσο την αναβάθμιση των ηλεκτρονικών εξαρτημάτων με την προσθήκη νέων συσκευών, όσο και τη χρήση κατάλληλων μεθοδολογικών εργαλείων.

Προτείνεται η περαιτέρω εξέλιξη του συστήματος με την ενσωμάτωση επιπλέον αισθητήρων, βηματικών κινητήρων και σερβοκινητήρων, οι οποίοι θα προσφέρουν νέες δυνατότητες. Για την ενσωμάτωσή τους, απαιτείται ο σχεδιασμός των κατάλληλων πλακετών και η προσθήκη νέων μπλοκ στο λογισμικό. Σχετικά με την ανάπτυξη του λογισμικού της πλατφόρμας, είναι σημαντικό ότι το ίδιο το λογισμικό μπορεί εύκολα να εμπλουτιστεί με επιπλέον δομικά στοιχεία, γιατί οι σημαντικές δυσκολίες στην ανάπτυξή του έχουν ήδη αντιμετωπιστεί. Με την ίδια φιλοσοφία ανάπτυξης, το λογισμικό μπορεί εύκολα να εμπλουτιστεί με επιπλέον ηλεκτρονικά δομικά στοιχεία που μπορούν να επεκτείνουν τις δυνατότητες της προτεινόμενης πλατφόρμας.

Επιπλέον, θα ήταν χρήσιμο να εξεταστεί η ενσωμάτωση τεχνολογιών εικονικής και/ή επαυξημένης πραγματικότητας, με σκοπό τη δημιουργία ενός πλήρους και καινοτόμου εργαλείου εκπαίδευσης.

Τέλος θα αποτελούσε σημαντική προσθήκη η δημιουργία συνοδευτικού εκπαιδευτικού υλικού ανά διδακτικό αντικείμενο, το οποίο θα μπορεί να αξιοποιηθεί από τους εκπαιδευτικούς ως σημείο αναφοράς. Οποιαδήποτε περαιτέρω ανάπτυξη της προτεινόμενης πλατφόρμας θα πρέπει να υποστηριχθεί και να πραγματοποιηθεί μέσω της συνεχούς αλληλεπίδρασης με μαθητές και εκπαιδευτικούς. Απώτερος στόχος της προτεινόμενης προσέγγισης είναι να αποτελέσει ένα ανοικτό εργαλείο που θα υποστηρίζεται από μια κοινότητα εκπαιδευτικών και μαθητών και θα εμπλουτίζεται συνεχώς από την αλληλεπίδρασή τους.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- 
1. de Vries M (2018) Handbook of Technology Education
  2. Ahmed Alanazi (2016) A Critical Review of Constructivist Theory and the Emergence of Constructionism. *American Research Journal of Humanities and Social Sciences*. <https://doi.org/10.21694/2378-7031.16018>
  3. Bers MU, Flannery L, Kazakoff ER, Sullivan A (2014) Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education* 72:. <https://doi.org/10.1016/j.compedu.2013.10.020>
  4. Kakaras G, Goumenakis E, Glynos E, et al (2022) HYDRA 2.0: Towards developing a holistic tool for STEM education. In: 2022 30th Mediterranean Conference on Control and Automation (MED). IEEE, pp 1099–1104
  5. Tsalmpouris G, Tsinarakis G, Gertsakis N, et al (2021) Hydra: Introducing a low-cost framework for stem education using open tools. *Electronics (Switzerland)* 10:. <https://doi.org/10.3390/electronics10243056>
  6. Alimisis D (2013) Educational robotics: Open questions and new challenges
  7. Eguchi A, Shen J (2013) Student Learning Experience through CoSpace Educational Robotics. pp 93–127
  8. Holmlund TD, Lesseig K, Slavitt D (2018) Making sense of “STEM education” in K-12 contexts. *International Journal of STEM Education* 5:32. <https://doi.org/10.1186/s40594-018-0127-2>
  9. Chin K-Y, Hong Z-W, Chen Y-L (2014) Impact of Using an Educational Robot-Based Learning System on Students’ Motivation in Elementary Education. *IEEE Transactions on Learning Technologies* 7:333–345. <https://doi.org/10.1109/TLT.2014.2346756>
  10. Atmatzidou S, Demetriadis S, Nika P (2018) How Does the Degree of Guidance Support Students’ Metacognitive and Problem Solving Skills in Educational Robotics? *Journal of Science Education and Technology* 27:70–85. <https://doi.org/10.1007/s10956-017-9709-x>
  11. McGill MM (2012) Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education* 12:. <https://doi.org/10.1145/2133797.2133801>
-

12. Evripidou S, Georgiou K, Doitsidis L, et al (2020) Educational Robotics: Platforms, Competitions and Expected Learning Outcomes. IEEE Access 8:219534–219562. <https://doi.org/10.1109/ACCESS.2020.3042555>
13. Resnick M, Maloney J, Monroy-Hernández A, et al (2009) Scratch. Communications of the ACM 52:60–67. <https://doi.org/10.1145/1592761.1592779>
14. Maloney J, Resnick M, Rusk N, et al (2010) The Scratch Programming Language and Environment. ACM Transactions on Computing Education 10:1–15. <https://doi.org/10.1145/1868358.1868363>
15. Ανακτήθηκε από <https://s4a.cat/>
16. Ανακτήθηκε από <https://snap.berkeley.edu/>
17. Ανακτήθηκε από <https://developers.google.com/blockly>
18. Evripidou S, Doitsidis L, Tsinarakis G, et al (2022) Selecting a Robotic Platform for Education. In: 2022 IEEE International Conference on Consumer Electronics (ICCE). IEEE, pp 1–6
19. Ανακτήθηκε από USB-C\_Datasheet
20. Ανακτήθηκε από Pinout-NANO\_Datasheet
21. Ανακτήθηκε από <https://www.kicad.org/>
22. Ανακτήθηκε από <http://flatcam.org/>
23. Ανακτήθηκε από Ardublockly's web page <https://ardublockly.embeddedlog.com/>
24. Ανακτήθηκε από <https://github.com/carlosperate/ardublockly>
25. Ανακτήθηκε από <https://github.com/carlosperate/ardublockly/wiki>
26. <https://www.printables.com/model/47787-printablok-source> Printablok system, accessed: 2024-02-19.
27. Abu Khurma O, Al Darayseh A, Alramamneh Y (2022) A Framework for Incorporating the “Learning How to Learn” Approach in Teaching STEM Education. Education Sciences (Basel) 13:1. <https://doi.org/10.3390/educsci13010001>
28. Kelley TR, Knowles JG (2016) A conceptual framework for integrated STEM education. International Journal of STEM Education 3:11. <https://doi.org/10.1186/s40594-016-0046-z>

## **Παράρτημα 1: Φύλλα Εργασίας και Ερωτηματολόγια**

---

Στο Παράρτημα 1 παρατίθενται τα φύλλα έργου και τα ερωτηματολόγια όπως αυτά δόθηκαν στους μαθητές. Πρώτο φύλλο έργου:

# Εργασία 1

---

## 1. Τι θέλουμε να πετύχουμε;

Να ανάβουν τα φωτάκια όταν πατάμε τα αντίστοιχα κουμπάκια.



## 2. Έχεις μπροστά σου τα παρακάτω εξαρτήματα;

LED



Διακόπτες



Μικροελεγκτής



Αντιστάσεις



Καλώδια



Πλακέτα δοκιμών



Ας δούμε λίγη θεωρία...







**LED:** Φωτάκια ή ποιο σωστά Δίοδος εκπομπής φωτός, είναι μια συσκευή η οποία μετατρέπει μια ποσότητα ηλεκτρικής ενέργειας σε «φως» .

**Αντιστάσεις:** Είναι ένα ηλεκτρονικό εξάρτημα που περιορίζει το ηλεκτρικό ρεύμα, το χρειάζονται τα LED για να ανάψουν. Είναι αχώριστοι φίλοι, πάντα τοποθετούνται μαζί.

**Διακόπτες:** Είναι κουμπάκια με τα οποία μπορούμε να διακόπτουμε ή να επιτρέπουμε τη ροή του ηλεκτρικού ρεύματος.

**Μικροελεγκτής:** Το όνομα του είναι «Arduino» και είναι ο εγκέφαλος ενός κυκλώματος. Ελέγχει όλα τα ηλεκτρονικά εξαρτήματα για την εκτέλεση συγκεκριμένων εργασιών.

**Καλώδια:** Είναι μονωμένος αγωγός, δηλαδή σύρμα καλυμμένο με πλαστικό, για την μεταφορά ηλεκτρικού ρεύματος.

**Πλακέτα δοκιμών:** Είναι μια βάση πάνω στην οποία τοποθετούμε ηλεκτρονικά εξαρτήματα ώστε να δοκιμάσουμε ένα ηλεκτρικό κύκλωμα.

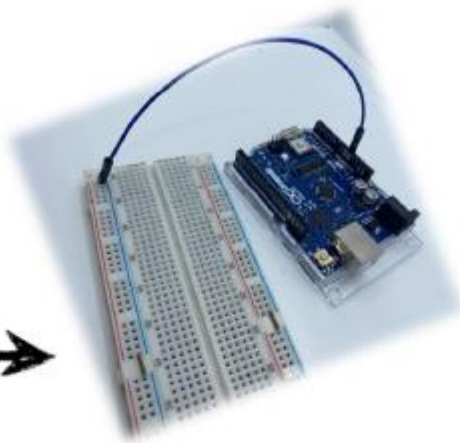
**Ηλεκτρικό ρεύμα:** Είναι η ροή των μικροσκοπικών σωματιδίων μέσα στα καλώδια, και αυτή η ροή μας βοηθάει να ανάβουμε τα φώτα, να βλέπουμε τηλεόραση και να χρησιμοποιούμε όλες τις ηλεκτρικές συσκευές στο σπίτι μας.

**Ηλεκτρικό κύκλωμα:** Είναι σαν μια κυκλική διαδρομή, ένας δρόμος που ξεκινά από μία πηγή ενέργειας (όπως η μπαταρία ή η πρίζα), περνάει μέσα από τα καλώδια και επιστρέφει πίσω στην πηγή. Για να λειτουργήσει όμως σωστά το κύκλωμα, πρέπει ο δρόμος να είναι συνεχής, χωρίς κενά.



### 3. Ώρα για πείραμα!

**1** Τοποθέτησε τον εγκεφαλο πάνω στην πλακέτα δοκιμών όπως φαίνεται στην εικόνα. Βάλε την μια άκρη του μπλε καλωδίου στο γράμμα GND του εγκεφάλου και την άλλη πάνω από την μπλε γραμμή στην πλακέτα δοκιμών.

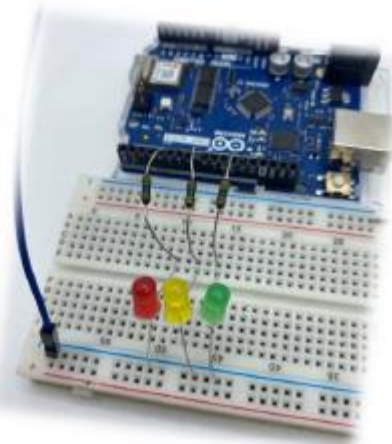


**2** Τοποθέτησε τις τρεις αντιστάσεις με το ένα ποδαράκι της κάθε μίας στα γράμματα D7,D8,D12 του εγκεφάλου. Τα άλλα ποδαράκια σε κάποιο διαφορετικό κυκλάκι της πλακέτας δοκιμών το ένα δίπλα στο άλλο.

Συμβουλέψου την εικόνα.

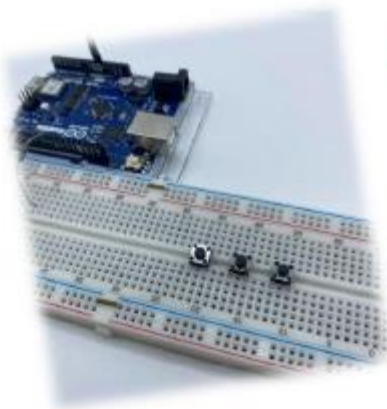
**3** Τώρα βάλε το μακρύ ποδαράκι από το κάθε φωτάκι κάτω από κάθε ποδαράκι της αντίστασης. Τέλος βάλε το κοντό ποδαράκι από τα φωτάκια κάτω από την μπλε γραμμή.

Ετοιμό; Να μοιάζει με την φωτογραφία



Πρόσεξε να μην ακουμπάνε τα ποδαράκια μεταξύ τους!

Ας προχωρήσουμε λοιπόν.



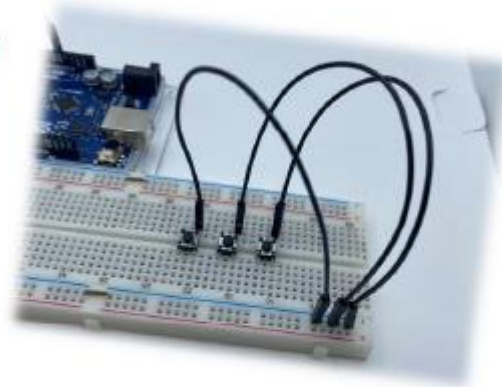
- 4 Τοποθέτησε τα κουμπιά όπως βλέπεις στην εικόνα.



- 5 Έπειτα βάλε τα μαύρα καλώδια όπως φαίνεται στην εικόνα.



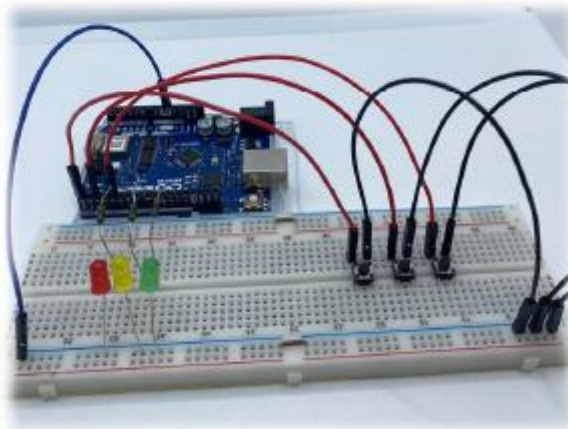
Η μια άκρη θα είναι στην αριστερή άκρη του κουμπιού και η άλλη κάτω από την μπλε γραμμή.



Έλα φτάσαμε!



Πάμε στο τελευταίο βήμα!



- 6 Τοποθέτησε τα κόκκινα καλώδια όπως φαίνεται στην φωτογραφία.



Η μια άκρη θα είναι στην δεξιά άκρη του κουμπιού και η άλλη στα γράμματα RX0,D2,D4 του εγκεφάλου.



Έτοιμο! Φώναξε τον διδάσκοντα να το δείξεις και πες του να το προγραμματίσετε!



## Εργασία 2

---

Ήρθε η ώρα να γνωρίσουμε το ColourBot !

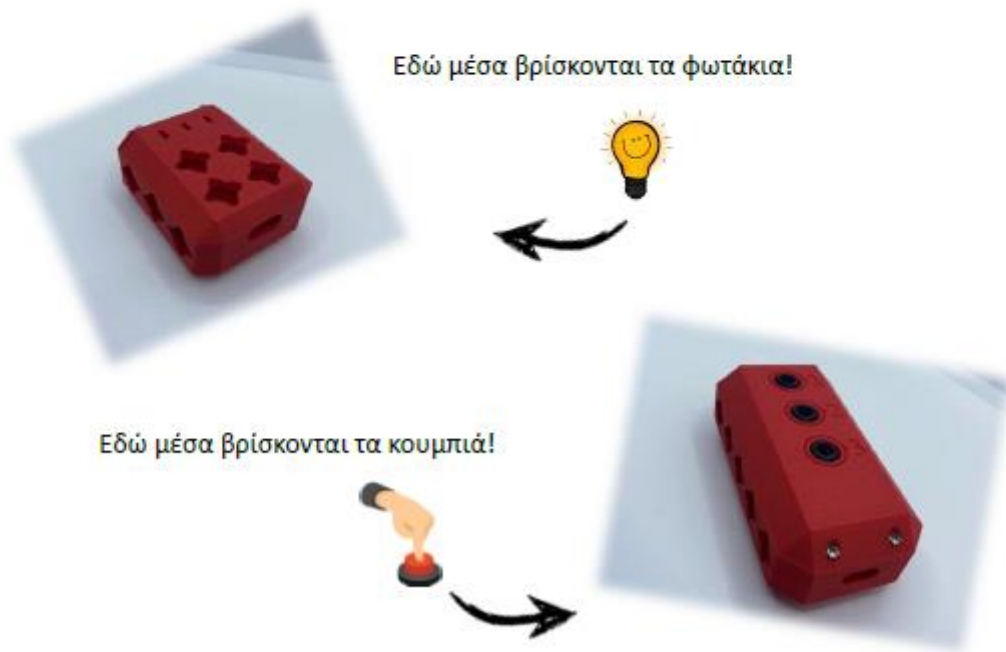
### 1. Τι θέλουμε να πετύχουμε;

Όπως και πριν! Να ανάβουν τα φωτάκια όταν πατάμε τα αντίστοιχα κουμπάκια.

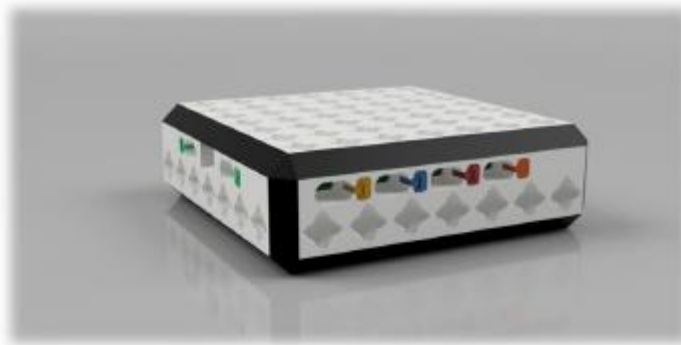


### 2. Γνωριμία με τα εξαρτήματα.

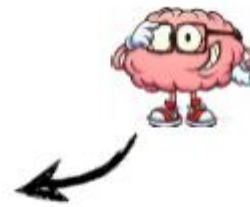
Τώρα όλα αυτά που είδαμε πριν είναι μέσα σε κουτάκια!







Και εδώ μέσα ο εγκέφαλος!



Θα χρειαστούμε ακόμα :



Καλώδια

&

Συνδέσεις



### 3. Ώρα για πείραμα!

Βήματα:

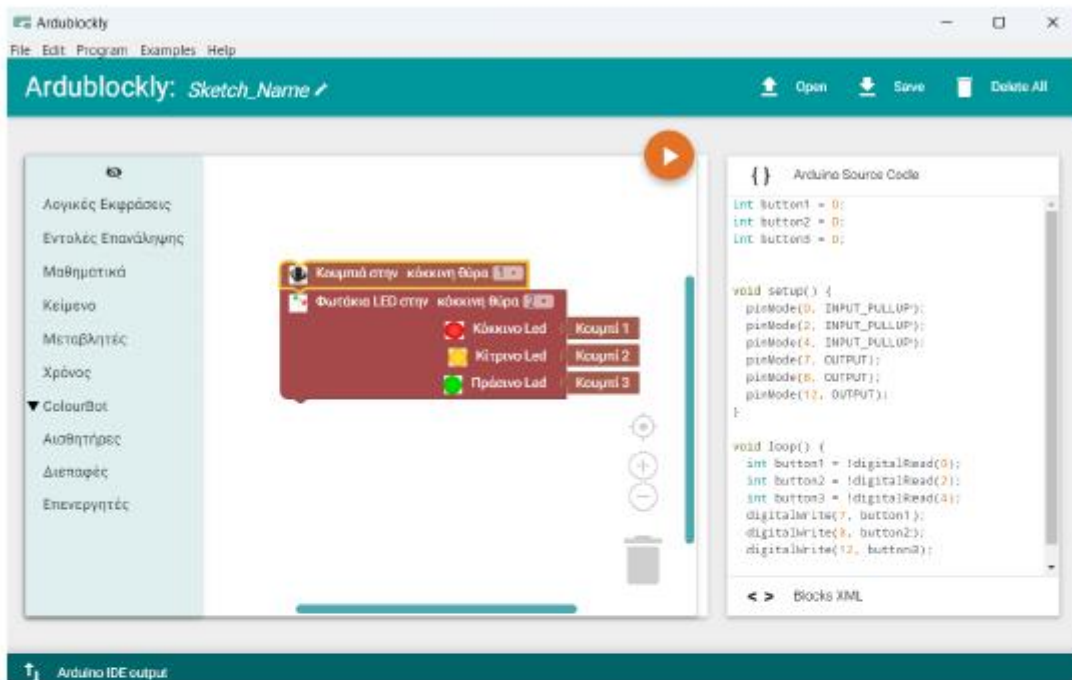


- 1 Σύνδεσε τα φωτάκια με ένα καλώδιο στο 2 του εγκεφάλου.
- 2 Σύνδεσε τα κουμπιά με ένα καλώδιο στο 1 του εγκεφάλου.
- 3 Αν θέλεις χρησιμοποίησε τις συνδέσεις για να τα ενώσεις μεταξύ τους.



**Έτοιμο! Φώναξε τον διδάσκοντα να το δείξεις  
και πες του να το προγραμματίσετε!**

4 Φτιάξε τον παρακάτω κώδικα.



**Συγχαρητήρια! Τα καταφέραμε!**



## Ερωτηματολόγιο

.....

1. Ποιο είναι το φύλο σου;

- ☐ Αγόρι  
☐ Κορίτσι

2. Πόσο ετών είσαι;

3. Με τι ασχολείσαι περισσότερο στον ελεύθερο σου χρόνο;

- ☐ Βιντεοπαιχνίδια (κονσόλα, υπολογιστής, smartphone, tablet)  
☐ Ανάγνωση εξωσχολικών βιβλίων  
☐ Βόλτες με φίλους  
☐ Αθλητισμός  
☐ Άλλο .....

4. Το περιεχόμενο του σημερινού μαθήματος ήταν ενδιαφέρον.

- |                       |                       |                               |                       |                       |
|-----------------------|-----------------------|-------------------------------|-----------------------|-----------------------|
| Διαφωνώ<br>απόλυτα    | Διαφωνώ               | Ούτε συμφωνώ/<br>ούτε διαφωνώ | Συμφωνώ               | Συμφωνώ<br>απόλυτα    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/>         | <input type="radio"/> | <input type="radio"/> |

5. Έχεις ξανακάνει πείραμα με ηλεκτρονικά εξαρτήματα;

- ☐ Ναι  
☐ Όχι

6. Μου άρεσε που συνεργάστηκα με τους συμμαθητές μου για το μάθημα.

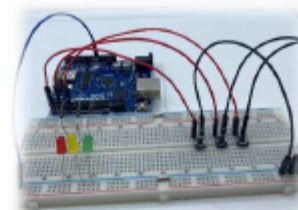
- |                       |                       |                               |                       |                       |
|-----------------------|-----------------------|-------------------------------|-----------------------|-----------------------|
| Διαφωνώ<br>απόλυτα    | Διαφωνώ               | Ούτε συμφωνώ/<br>ούτε διαφωνώ | Συμφωνώ               | Συμφωνώ<br>απόλυτα    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/>         | <input type="radio"/> | <input type="radio"/> |

7. Ανυπομονώ να κάνω κι άλλα παρόμοια (ή και πιο σύνθετα) πειράματα.

- |                       |                       |                               |                       |                       |
|-----------------------|-----------------------|-------------------------------|-----------------------|-----------------------|
| Διαφωνώ<br>απόλυτα    | Διαφωνώ               | Ούτε συμφωνώ/<br>ούτε διαφωνώ | Συμφωνώ               | Συμφωνώ<br>απόλυτα    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/>         | <input type="radio"/> | <input type="radio"/> |

**8. Η εργασία 1 όπου έφτιαξα κύκλωμα δεν με δυσκόλεψε.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



**9. Η εργασία 1 όπου έφτιαξα κύκλωμα δεν ήταν βαρετή.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**10. Θα ήθελα να ξανά κάνω πείραμα σαν την εργασία 1 όπου έφτιαξα κύκλωμα.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**11. Όταν αντίκρισα για πρώτη φορά το κύκλωμα μου φάνηκε ενδιαφέρον.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**12. Η εργασία 2 με το ColourBot δεν με δυσκόλεψε.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



**13. Η εργασία 2 με το ColourBot δεν ήταν βαρετή.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**14. Θα ήθελα να ξανά κάνω πείραμα σαν την εργασία 2 με το ColourBot.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**15. Όταν αντίκρισα για πρώτη φορά το ColourBot μου φάνηκε ενδιαφέρον**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**16. Τι σου άρεσε περισσότερο και τι όχι στο σημερινό μάθημα;**

.....

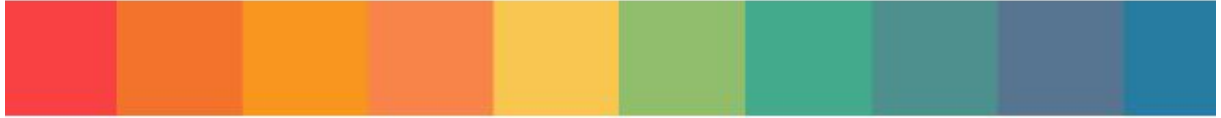
.....

.....

Δεύτερο φύλλο έργου:



## Φύλλο Εργασίας



Σήμερα θα παίζουμε και θα μάθουμε μαζί με το **COLOURBOT**  
Ας δούμε από τι αποτελείται.



Αυτό το κουτί είναι ο εγκέφαλος του. Χωρίς αυτό δεν μπορεί να λειτουργήσει.



Από αυτό το κουτί παίρνει τροφοδοσία (ρεύμα) για τα λειτουργήσει.



Εδώ μέσα βρίσκονται 3 LED (φωτάκια).



Εδώ μέσα βρίσκονται 3 διακόπτες.



Εδώ μέσα βρίσκεται ένας αισθητήρας απόστασης.



Εδώ μέσα βρίσκεται ένα τηλεχειριστήριο.



Εδώ μέσα βρίσκεται ένας κινητήρας που συνδέεται με μια ρόδα.



Εδώ μέσα βρίσκεται μια οθόνη.





Εδώ μέσα βρίσκεται ένας αισθητήρας θερμοκρασίας και υγρασίας.



Εδώ μέσα βρίσκεται ένας αισθητήρας ανίχνευσης εμποδίου.



Εδώ μέσα βρίσκεται ένας αισθητήρας ανίχνευσης γραμμής.

- Ανάλογα το χρώμα του κουτιού μπορούμε να το συνδέσουμε στο αντίστοιχο χρώμα στον εγκέφαλο. Για παράδειγμα τα δυο κόκκινα συνδέονται μόνο στο κόκκινο  και στο κόκκινο .
- Όποιο κουτί θέλουμε να χρησιμοποιήσουμε αναγκαστικά το συνδέουμε με τα καλώδια στον εγκέφαλο.
- Πρέπει πάντα να τα προγραμματίζουμε όπως μάθαμε στην παρουσίαση.

Αυτά είναι τα καλώδια μας :

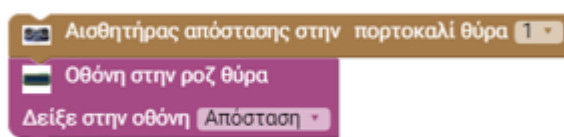


Και αυτοί είναι οι σύνδεσμοι για να ενώνουμε τα κουτιά όπως μας αρέσει :





Ανάλογα ποιο  
διακόπτη πατάμε  
ανάβουν  
διαφορετικά τα LED



Εμφάνισε στην  
οθόνη την  
απόσταση από  
κάποιο εμπόδιο



Αν υπάρχει εμπόδιο  
ανάβουν τα LED

## **Ήρθε η ώρα να φτιάξουμε το δικό μας ρομπότ!**

### Θέλουμε να:

- μπορεί να κινείται
- να βλέπει εμπόδια και να σταματάει ή να ελαττώνει ταχύτητα πριν σταματήσει
- να έχει φωτεινή ένδειξη όταν πηγαίνει κοντά σε εμπόδιο-σταματήσει
- να βλέπουμε τι αποστάσεις βλέπει
- να κάνει όπισθεν και να σταματάει αν πίσω του έχει εμπόδιο.

## Ερωτηματολόγιο



1. Έχεις ξανά χρησιμοποιήσει κάτι παρόμοιο;

☐ Ναι

☐ Όχι

2. Το περιεχόμενο του σημερινού μαθήματος ήταν ενδιαφέρον.

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐

3. Όταν αντίκρισα για πρώτη φορά το ColourBot μου φάνηκε ενδιαφέρον

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐

4. Ανυπομονώ να κάνω κι άλλα παρόμοια (ή και πιο σύνθετα) πειράματα.

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐

5. Σε σύγκριση με άλλα συστήματα που έχω χρησιμοποιήσει αυτό μου άρεσε περισσότερο.

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐

6. Μου άρεσε να ενώνω με τα χέρια μου τα κουτιά του ColourBot.

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐

7. Μου φάνηκε εύκολος ο προγραμματισμός του ColourBot .

Διαφωνώ  
απόλυτα

Διαφωνώ

Ούτε συμφωνώ/  
ούτε διαφωνώ

Συμφωνώ

Συμφωνώ  
απόλυτα

☐☐☐☐☐



Οι προτάσεις που ακολουθούν, αναφέρονται στο πώς αισθάνθηκε κατά τη διάρκεια της προηγούμενης δραστηριότητας.

**1. Η προσπάθεια μου έγινε εύκολα, χωρίς δυσκολίες.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**2. Δεν κατάλαβα ποτέ πέρασε η ώρα.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**3. Δε δυσκολεύτηκα να συγκεντρωθώ..**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**4. Η σκέψη μου ήταν ξεκάθαρη.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**5. Ήμουν εντελώς απορροφημένος σ' αυτό που έκανα.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**6. Οι σωστές λύσεις έρχονταν στο μυαλό από μόνες τους.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**7. Ήξερα τι να κάνω σε κάθε βήμα.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**8. Αισθάνθηκα ότι είχα τον έλεγχο σε κάθε λεπτομέρεια.**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**9. Ήμουν απορροφημένος στις σκέψεις μου..**

Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ/ ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>