



Σεϊτανίδης Γεώργιος, Σχεδιασμός και σύνταξη αναλυτικών οδηγιών
χρήσης για Arduino και Micro:bit

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**



Πρόγραμμα Μεταπτυχιακών Σπουδών

«Τεχνολογίες και Υπηρεσίες Ευφών Συστημάτων Πληροφορικής και Επικοινωνιών



Διπλωματική Εργασία

Θέμα: Σχεδιασμός και σύνταξη αναλυτικών οδηγιών χρήσης για
Arduino και Micro:bit

Μεταπτυχιακός Φοιτητής: Σεϊτανίδης Γεώργιος

ΑΜ: 9093202001021

Επιβλέπον Καθηγητής: Κος Χριστοδούλου Σωτήριος

Πάτρα, Δεκέμβριος 2022

Διπλωματική Εργασία





Σεϊτανίδης Γεώργιος, Σχεδιασμός και σύνταξη αναλυτικών οδηγιών
χρήσης για Arduino και Micro:bit

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**



Πρόγραμμα Μεταπτυχιακών Σπουδών

«Τεχνολογίες και Υπηρεσίες Ευφρών Συστημάτων Πληροφορικής και Επικοινωνιών



3

**ΘΕΜΑ: : Σχεδιασμός και σύνταξη αναλυτικών οδηγιών χρήσης
για Arduino και Micro:bit**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΕΪΤΑΝΙΔΗΣ ΓΕΩΡΓΙΟΣ

Επιβλέπων Καθηγητής:
Χριστοδούλου Σωτήριος

Μέλος Επιτροπής:

ΠΑΤΡΑ, Δεκέμβριος 2022

Διπλωματική Εργασία





ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και οι πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».



ΕΥΧΑΡΙΣΤΙΕΣ

Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Χριστοδούλου Σωτήριο, ο οποίος ήταν συνεχώς δίπλα μου και με τις κατευθυντήριες γραμμές που μου έδωσε κατάφερα να την εκπονήσω.

Τέλος , αφιερώνω την διπλωματική μου εργασία στον πατέρα μου, ο οποίος δεν βρίσκεται πλέον εν ζωή, την οικογένεια μου, την κοπέλα μου, τους φίλους μου καθώς και τον κόουτς μου οι οποίοι με στήριξαν ο καθένας με το δικό του τρόπο σε κάθε στιγμή της ακαδημαϊκής μου πορείας.



ΠΕΡΙΛΗΨΗ

Στις ημέρες μας η τεχνολογία έχει εισχωρήσει στις ζωές όλων μας. Τα STEM(Επιστήμη , Τεχνολογία , Μηχανική και Μαθηματικά) δειλά δειλά εισέρχονται στην εκπαίδευση με αρκετά πλεονεκτήματα, αλλά και με πολλά ζητήματα τα οποία είναι επιτακτική ανάγκη να λυθούν. Ένα εκπαιδευτικό εργαλείο το οποίο θεωρείται αρκετά ελπιδοφόρο είναι η ρομποτική. Στην εποχή μας τα εργαλεία της ρομποτικής εκπαίδευσης έχουν γνωρίσει μεγάλη άνθιση, έχοντας σκοπό την ανάδειξη της καινοτομίας καθώς και την ενεργοποίηση των μαθητών κατά τη διάρκεια του μαθήματος.

Στην συγκεκριμένη διπλωματική εργασία έγινε μελέτη σε εκπαιδευτικές θεωρίες και τεχνικές που χρησιμοποιούν την εκπαιδευτική ρομποτική. Στη συνέχεια υλοποιήθηκε οδηγός χρήσης για αρχαρίους στον μικροελεγκτή Micro:bit αλλά και στο Arduino Uno R3. Οι οδηγοί είναι ειδικά σχεδιασμένοι για μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης με την βοήθεια των πλατφορμών Make a code και Tinkercad. Τέλος πραγματοποιήθηκε επίδειξη δυνατοτήτων των πλατφορμών, έγιναν μικρές στοχευμένες εφαρμογές και πιο πολύπλοκα διδακτικά σενάρια.

7

Λέξεις - Κλειδιά

Εκπαιδευτική Ρομποτική, μικροελεγκτής Micro:bit, μικροελεγκτής Arduino, Οδηγοί χρήσης, Διδακτικά σενάρια.



Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	7
Περιεχόμενα.....	8
Κατάλογος εικόνων/ Σχημάτων.....	11
1.Εισαγωγή.....	13
1.1 Αντικείμενο της διπλωματικής.....	13
1.2 Μεθοδολογία	14
1.3 Σκοπός και στόχοι της εργασίας.....	14
1.4 Δομή και διάρθρωση της διπλωματικής εργασίας.....	14
2.Εκπαιδευτική Ρομποτική (STEM)	16
2.1 Εισαγωγή	16
2.2 Η έννοια της ρομποτικής.....	16
2.3 Η Έννοια της εκπαιδευτικής ρομποτικής	17
2.4 Οι προσεγγίσεις της εκπαιδευτικής ρομποτικής	18
2.4.1 Η εκπαιδευτική ρομποτική ως γνωστικό αντικείμενο	19
2.4.2 Η εκπαιδευτική ρομποτική ως εργαλείο εκπαίδευσης.....	19
2.5 Η εκπαιδευτική ρομποτική σε συνάρτηση με τα STEM	19
2.6 Συμπεράσματα	21
3. Ο μικροελεγκτής Arduino	21
3.1 Ιστορικά στοιχεία του Arduino.....	21
3.2 Οδηγός εκμάθησης Arduino.....	22
3.2.1 Τα περιβάλλοντα προγραμματισμού του Arduino και ο μικροελεγκτής.....	22
3.2.2 Απαραίτητα υλικά για την χρήση του Arduino	23
3.2.3 Εγκατάσταση του περιβάλλοντος Arduino IDE	24
3.2.4 Το ρεύμα λειτουργίας του Arduino.....	25
3.2.5 Τα Pins του Arduino (Είσοδοι/Εξοδοι)	26
3.2.6 Βασικές λειτουργίες του Arduino Uno R3 και προγραμματισμός	27
3.2.7 Η δομή επιλογής στο Arduino (If)	36
3.2.8 Η δομή επανάληψης στο Arduino (For-While)	37
3.2.9 Η χρήση του Breadboard στο Arduino	38



3.2.10 Τα κυριότερα εξαρτήματα που θα χρησιμοποιήσει ένας αρχάριος.....	40
3.3 Φύλλα εργασίας για αρχάριους στο Arduino	48
3.3.1 Φύλλο εργασίας 1- Το Led που αναβοσβήνει (Blink)	48
3.3.2 Φύλλο εργασίας 2- Η σταδιακή αύξηση και μείωση της φωτεινότητας στο Led (Fade in – Fade out).....	50
3.3.3 Φύλλο εργασίας 3 – Αυξομείωση της φωτεινότητας με αναβόσβημα στις άκρες (FadeIn – FadeOut με Blink στα Peaks)	52
3.3.4 Φύλλο εργασίας 4 – Κλιμακωτή αύξηση και μείωση της φωτεινότητας με ποτενσιόμετρο.....	54
3.3.5 Φύλλο εργασίας 5 – Η χρήση της σειριακής οθόνης στο Arduino	56
3.3.6 Φύλλο Εργασίας 6 – Χρησιμοποιώντας το RGB Led	58
3.3.7 Φύλλο εργασίας 7 – Τα φανάρια κυκλοφορίας μαζί με φανάρι πεζών με την βοήθεια των Led.....	61
3.3.8 Φύλλο εργασίας 8 – Τα φανάρια κυκλοφορίας με το φανάρι πεζών και με button διακοπής.....	63
3.3.9 Φύλλο Εργασίας 9 – Ανιχνεύοντας το φως με την βοήθεια φωτοαντίστασης.....	67
3.3.10 Φύλλο εργασίας 10 – Χρησιμοποιώντας τον ήχο με Piezo.....	69
3.3.11 Φύλλο εργασίας 11 – Ήχος με Piezo και ανίχνευση φωτός με φωτοαντίσταση 70	
3.3.12 Φύλλο εργασίας 12 – Χρησιμοποιώντας τον αισθητήρα για την μέτρηση απόστασης (HC-SR04).....	72
3.4 Διδακτικά σενάρια με την πλατφόρμα του Arduino.....	75
3.4.1 Το πρώτο διδακτικό σενάριο	75
3.4.2 Το δεύτερο διδακτικό σενάριο.....	91
4.0 μικροελεγκτής Micro:bit	104
4.1 Ιστορικά στοιχεία του μικροελεγκτή Micro:bit	104
4.2 Γενικές πληροφορίες και οδηγίες χρήσης για το micro:bit	105
4.3 Οδηγός εκμάθησης Micro:bit με φύλλα εργασίας	112
4.3.1 Φύλλο εργασίας 1 (Blink a led)	112
4.3.2 Φύλλο εργασίας 2 Το χαρούμενο και το λυπημένο Emoji	116
4.3.3 Φύλλο εργασίας 3 Εναλλαγή τυχαίων αριθμών	118
4.3.4 Φύλλο εργασίας 4 Εμφάνιση μηνυμάτων στην οθόνη του Micro:bit	120
4.3.5 Φύλλο εργασίας 5 Παιδικό τραγούδι με την βοήθεια του Micro:bit	123
5. Συμπεράσματα	126
5.1 Επίλογος	126
5.2 Συμπεράσματα	126



5.3 Βιβλιογραφία..... 129



Κατάλογος εικόνων/ Σχημάτων

Εικόνα 1-Περιβάλλον προγραμματισμού Arduino IDE	25
Εικόνα 2- Παράδειγμα Serial.print στην πλατφόρμα Tinkercad.....	36
Εικόνα 3- Παράδειγμα Breadboard.....	39
Εικόνα 4-Παράδειγμα συνδεσμολογίας με την βοήθεια του Breadboard	40
Εικόνα 5- Παράδειγμα από Leds στο Tinkercad.....	41
Εικόνα 6- Παράδειγμα RGB στην πλατφόρμα Tinkercad	42
Εικόνα 7- Παράδειγμα Pull down resistor στο Tinkercad	42
Εικόνα 8- Παράδειγμα Pull up resistor στο Tinkercad	43
Εικόνα 9- Παράδειγμα με ποτενσιόμετρο στο Tinkercad	44
Εικόνα 10- Παράδειγμα με ηχείο (Piezo) στο Tinkercad.....	45
Εικόνα 11- Παράδειγμα φωτοαντίστασης στο Tinkercad.....	46
Εικόνα 12- Παράδειγμα αισθητήρα υπερήχων στο Tinkercad	47
Εικόνα 13- Παράδειγμα Led Blink στο Tinkercad.....	48
Εικόνα 14- Παράδειγμα FadeIn - FadeOut στο Tinkercad	50
Εικόνα 15- Παράδειγμα FadeIn - FadeOut Blink at Peaks στο Tinkercad	52
Εικόνα 16- Παράδειγμα κυκλώματος με την σύνδεση ποτενσιόμετρου στο Tinkercad	55
Εικόνα 17 - Παράδειγμα σειριακής οθόνης στο Tinkercad.....	57
Εικόνα 18 - Παράδειγμα RGB Led στο Tinkercad.....	59
Εικόνα 19 - Παράδειγμα Φαναριού κυκλοφορίας με φανάρι πεζών στο Tinkercad.....	61
Εικόνα 20 - Παράδειγμα φαναριού κυκλοφορίας με φανάρι πεζών και κουμπί διακοπής στο Tinkercad.....	64
Εικόνα 21 - Παράδειγμα κυκλώματος με την βοήθεια φωτοαντίστασης στο Tinkercad	67
Εικόνα 22 - Παράδειγμα κυκλώματος Piezo στο Tinkercad.....	69
Εικόνα 23 - Παράδειγμα κυκλώματος Piezo με την χρήση φωτοαντίστασης στο Tinkercad .	71
Εικόνα 24 - Παράδειγμα κυκλώματος με αισθητήρα υπερήχων (HC-SR04) στο Tinkercad ...	72
Εικόνα 25 - Περιβάλλον λογισμικού Arduino	82
Εικόνα 26 - Παιχνίδι μνήμης στην πλατφόρμα του Tinkercad.....	85
Εικόνα 27 - Περιβάλλον λογισμικού Arduino2	98
Εικόνα 28 - Παράδειγμα αισθητήρα θερμοκρασίας LM.....	101
Εικόνα 29 - Παράδειγμα αισθητήρα LM	104
Εικόνα 30 - Παράδειγμα Micro:bit στο Make a code.....	106
Εικόνα 31 - Τα εξαρτήματα του Micro:bit.....	107
Εικόνα 32 - Παράδειγμα LEDs στο Micro:bit.....	108
Εικόνα 33 - Παράδειγμα κουμπιών στο Micro:bit	108
Εικόνα 34 - Παράδειγμα των Pins στο Micro:bit.....	109
Εικόνα 35 - Παράδειγμα αισθητήρα φωτός στο Micro:bit	109
Εικόνα 36 - Παράδειγμα αισθητήρα θερμοκρασίας στο Micro:bit	109
Εικόνα 37 - Παράδειγμα του επιταχυνσιόμετρου στο Micro:bit	110
Εικόνα 38 - Παράδειγμα πυξίδας στο Micro:bit	110
Εικόνα 39 - Παράδειγμα ραδιοεπικοινωνίας στο Micro:bit	111
Εικόνα 40 - Παράδειγμα Bluetooth στο Micro:bit	111
Εικόνα 41 - Παράδειγμα θύρας Micro-USB στο Micro:bit	112
Εικόνα 42 - Παράδειγμα προσωμοιωτή Micro:bit.....	114



Εικόνα 43 - Παράδειγμα Blink a led στο Micro:bit..... 114
Εικόνα 44 - Παράδειγμα Sad η Happy Emoji στο Micro: bit 117
Εικόνα 45 - Παράδειγμα εναλλαγής τυχαίων αριθμών στο Micro:bit..... 119
Εικόνα 46 - Παράδειγμα εμφάνισης μηνυμάτων και συμβολοσειρών στο Micro:bit..... 122
Εικόνα 47 - Παράδειγμα παιδικού τραγουδιού στο Micro:bit 124



1.Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Στις ημέρες μας είναι αλήθεια ότι η εξέλιξη της τεχνολογίας είναι τεράστια. Η ρομποτική τεχνολογία καθώς και οι αυτοματισμοί έχουν εισχωρήσει στις ζωές των ανθρώπων, με αποτέλεσμα να τις συναντάμε καθημερινά στη ζωή μας. Οι νέοι σήμερα αναπτύσσονται και μεγαλώνουν μαζί με την τεχνολογία, χρησιμοποιώντας παράλληλα αυτοματοποιημένα συστήματα, έξυπνες συσκευές και ρομποτικές κατασκευές. Ωστόσο αν και ζουν μέσα σ' αυτά οι μαθητές σχεδόν ποτέ δεν τους γεννιέται η απορία πως λειτουργούν και από που έχουν ξεκινήσει. Με τον κίνδυνο στο μέλλον να καταλήξουμε να έχουμε παθητικούς χρήστες των εν λόγω συσκευών. Μέσω των ερωτήσεων της περιέργειας αλλά και της αναζήτησης για το πως λειτουργούν αλλά και το πως επισκευάζονται όλα αυτά έχουν την δυνατότητα να πάνε ένα βήμα παρακάτω. Η εκπαιδευτική ρομποτική μας δίνει την ευκαιρία οι μαθητές να κατασκευάσουν να αναζητήσουν, καθώς και να σχεδιάσουν τέτοια συστήματα δίνοντας τους την δυνατότητα να μάθουν με ένα πιο διαδραστικό τρόπο. Αντικείμενο αυτής την διπλωματικής εργασίας είναι η χρήση και η εκμάθηση των μικροελεγχτών Microbit και Arduino στην εκπαιδευτική ρομποτική. Το Microbit και το Arduino θεωρούνται ιδανικά για χρήση στην εκπαίδευση, διότι και τα δύο είναι πλακέτες ανοιχτού κώδικα, το κόστος τους δεν είναι απαγορευτικό και έχουν τεράστια συμβατότητα σε αισθητήρες. Επίσης προσφέρουν πολλά αναπτυξιακά περιβάλλοντα για τον προγραμματισμό των μικροελεγκτών. Αρκετά λειτουργούν με τον οπτικό προγραμματισμό και άλλα με κώδικα. Δίνοντας την δυνατότητα στην τεχνολογία Microbit και Arduino να είναι ευέλικτες να και μπορούν να προσαρμοστούν σ' όλες τις βαθμίδες της εκπαίδευσης. Χρησιμοποιώντας τις ενδεδειγμένες παιδαγωγικές θεωρίες και τεχνικές στην εκπαιδευτική διαδικασία, η εκπαιδευτική ρομποτική παρουσιάζει θετικό αντίκτυπο στους μαθητές όχι μόνο σε γνωστικό επίπεδο αλλά και σε συναισθηματικό και στο κοινωνικό.



1.2 Μεθοδολογία

Στο πλαίσιο της συγκεκριμένης διπλωματικής εργασίας έγινε εκμάθηση της πλατφόρμας του Microbit, καθώς και του Arduino uno. Η εκμάθηση πραγματοποιήθηκε στο Make a code για το Microbit και για το Arduino χρησιμοποιήθηκε η πλατφόρμα του Tinkercad. Εν συνεχεία έγινε μελέτη βιβλιογραφίας σε παιδαγωγικές θεωρίες και τεχνικές που βασίζονται στην εκπαιδευτική ρομποτική, έγινε χρήση της μεθοδολογίας της ανάπτυξης των διδακτικών σεναρίων. Ακολούθησε η συγγραφή οδηγού εκμάθησης για αρχαρίους Microbit καθώς και Arduino, ο οποίος περιλαμβάνει την Ε' τάξη του δημοτικού και καταλήγει έως και την Γ' λύκειού. Τέλος δημιουργήθηκαν εκπαιδευτικά σεναρία για κάποιες τάξεις όπου η δυσκολία των σεναρίων κλιμακώνεται με το πέρασμα σε κάθε επόμενη τάξη.

14

1.3 Σκοπός και στόχοι της εργασίας

Ο σκοπός της διπλωματικής εργασίας είναι η εκμάθηση και ανάδειξη της εκπαιδευτικής ρομποτικής χρησιμοποιώντας τους μικροελεγκτές Microbit και Arduino. Στόχοι είναι η μελέτη των πλατφορμών χαμηλού κόστους για ανάπτυξη STEM εφαρμογών Microbit και Arduino. Θα πραγματοποιηθεί οδηγός χρήσης για τις δύο πλατφόρμες ειδικά σχεδιασμένος για μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης. Επιπλέον θα υλοποιηθεί επίδειξη των δυνατοτήτων των πλατφορμών, με μικρές εφαρμογές και θα ακολουθούν ενδεικτικά διδακτικά σεναρία για κάθε τάξη.

1.4 Δομή και διάρθρωση της διπλωματικής εργασίας

Η διπλωματική εργασία δομείται σε πέντε κεφάλαια και περιέχει δύο οδηγούς εκμάθησης καθώς και διδακτικά σεναρία.



Στο πρώτο κεφάλαιο παρουσιάζεται το αντικείμενο που ασχολείται η διπλωματική εργασία. Ακολουθεί η μεθοδολογία την οποία επιλέξαμε και οι στόχοι που έχουμε θέσει. Οι στόχοι που έχουμε θέσει προσπαθήσαμε να είναι ρεαλιστικοί και να βασίζονται στην ελληνική πρωτοβάθμια και δευτεροβάθμια εκπαίδευση. Φροντίσαμε η δυσκολία να είναι κλιμακωτή για να μπορούν να ανταποκριθούν όλοι οι εκπαιδευόμενοι.

Στο δεύτερο κεφάλαιο αναφερθήκαμε για την ρομποτική, καθώς και για την εκπαιδευτική ρομποτική σε θεωρητικό επίπεδο. Προσεγγίσαμε την εκπαιδευτική ρομποτική ως έννοια και είδαμε το πως λειτουργεί ως εκπαιδευτικό εργαλείο, αλλά και το πως είναι σε συνάρτηση με τα STEM.

Συνεχίζοντας στο τρίτο κεφάλαιο πραγματοποιήσαμε έναν οδηγό εκμάθησης Arduino. Οι τάξεις στις οποίες απευθύνετε είναι από την Τρίτη γυμνασίου έως και το τέλος της τρίτης λυκείου. Το κεφάλαιο ολοκληρώνεται με δύο διδακτικά σενάρια τα οποία είναι για προχωρημένους μαθητές της τρίτης λυκείου.

Στο τέταρτο κεφάλαιο κάνουμε εισαγωγή στο Micro:bit. Το κεφάλαιο αυτό αναφέρεται σε φύλλα εργασίας, τα οποία αφορούν την Ε' δημοτικού έως και την Β γυμνασίου. Στο συγκεκριμένο κεφάλαιο πλην των φύλλων εργασίας, θεωρήθηκε σωστό να μην υλοποιηθούν εκπαιδευτικά διότι οι εκπαιδευόμενοι στην ηλικία αυτή δεν θα ανταποκρίνονταν στην δυσκολία τους.

Τέλος στο πέμπτο κεφάλαιο η διπλωματική εργασία κλείνει με τον επίλογο και τα συμπεράσματα τα οποία βγάλαμε. Ολοκληρώνοντας τοποθετείται η βιβλιογραφία την οποία χρησιμοποιήσαμε για να υλοποιήσουμε την συγκεκριμένη διπλωματική.



2. Εκπαιδευτική Ρομποτική (STEM)

2.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο θα γίνει παρουσίαση της έννοιας της εκπαιδευτικής ρομποτικής καθώς και των ποικίλων προσεγγίσεών της. Θα υλοποιηθεί παρουσίαση των οφελών της εκπαιδευτικής ρομποτικής στα πλαίσια του STEM.

2.2 Η έννοια της ρομποτικής

Το ρομπότ θεωρείται μια μηχανική συσκευή που έχει την δυνατότητα να υποκαθιστά τον ανθρώπινο παράγοντα σε επικίνδυνες και δύσκολες εργασίες. Η χρήση του ρομπότ μπορεί να πραγματοποιηθεί είτε μέσω του ανθρώπου ή να λειτουργήσει αυτόνομα έπειτα από προγραμματισμό ενός υπολογιστή. Η ετυμολογία της λέξης προέρχεται από το σλάβικο (Robota) το οποίο μεταφράζεται ως εργασία. Η καθιέρωση της ορολογίας έγινε το 1920 από τον Τσέχο θεατρικό συγγραφέα Karel Capek, ο οποίος στο έργο του RUR (Rossum's Universal Robots) σατιρίζει το πόσο εξαρτημένη είναι η κοινωνία από τα ρομπότ και που τελικά εξοντώνουν αυτούς που τους έχουν δημιουργήσει. Σήμερα σε αρκετές σλάβικες γλώσσες η λέξη χρησιμοποιείται και μεταφράζεται με την έννοια της σκληρής δουλειάς. Ως ρομποτική θεωρείται ο τεχνολογικός κλάδος που αντικείμενο του είναι η έρευνα, ο σχεδιασμός και η λειτουργία των ρομπότ. Τα τελευταία χρόνια η ανάπτυξη της ρομποτικής είναι τεράστια, έχοντας συνέπεια τη συνεχώς αυξανόμενη χρήση των ρομπότ σε αρκετούς τομείς. Μερικοί απ' αυτούς είναι οι βιομηχανίες οι οποίες πλέον είναι αναγκασμένες να τη χρησιμοποιούν. Η ρομποτική διευκολύνει αρκετά σημεία της παραγωγικής διαδικασίας, καθώς βελτιώνει την ποιότητα των προϊόντων και αυξάνει την παραγωγικότητα. Επιπλέον προστατεύει τον άνθρωπο από αρκετές εργασίες οι οποίες θα ήταν δύσκολες και ανθυγιεινές σε μία βιομηχανία.



2.3 Η Έννοια της εκπαιδευτικής ρομποτικής

Ως εκπαιδευτική ρομποτική νοείται ένα προοδευτικό εργαλείο που αφορά την εκπαίδευση, δίνοντας στους μαθητές τη δυνατότητα κατασκευής και ελέγχου αντικειμένων από την καθημερινή ζωή κάνοντας χρήση προγραμματισμού. Μέσω του αντικειμένου αυτού είναι εφικτό να διδαχτούν ορισμένα αντικείμενα που άπτονται των τομέων των θετικών επιστημών, της ηλεκτρονικής, των μαθηματικών, της φυσικής και πρωτίστως της πληροφορικής. Με την πάροδο των χρόνων η εκπαιδευτική ρομποτική έχει εισχωρήσει στις σχολικές μονάδες. Αυτό έχει ως συνέπεια την εξέλιξη σημαντικών δεξιοτήτων των μαθητών όπως η ομαδικότητα, η δημιουργικότητα και η συνεργασία. Μέσω της εκπαιδευτικής ρομποτικής ενεργοποιείται η δημιουργικότητα των μαθητών και υπάρχει η πιθανότητα μέσα από αυτή αρκετοί μαθητές να εκφράζονται. Οι μαθητές αναλαμβάνουν τον ρόλο μηχανικού καθώς κάνουν χρήση και παίζουν με κινητήρες και αισθητήρες. Επιπλέον δημιουργούν και διαμοιράζουν τα δικά τους έργα αναλαμβάνοντας και τον ρόλο του αφηγητή, καθώς τα έργα τους αποκτούν διάδραση με το περιβάλλον τους.

Συμπερασματικά η εκπαιδευτική ρομποτική περιέχει:

- ✚ Φυσικά προγραμματιζόμενα αντικείμενα τα οποία θεωρούνται απλά αντικείμενα που με τα ενδεδειγμένα εξαρτήματα υπάρχει η δυνατότητα να πραγματοποιηθεί μια πολυπλοκότερη κατασκευή, με την προοπτική επεξεργασίας πληροφοριών για την λήψη αποφάσεων, χρησιμοποιώντας αισθητήρια και κινητήρες.
- ✚ Το προγραμματιστικό περιβάλλον από το οποίο προγραμματίζονται τα φυσικά αντικείμενα. Η πληροφορία λειτουργεί ως είσοδος στον προγραμματισμό με την βοήθεια των αισθητήρων, γίνεται επεξεργασία της και εξάγει αποτέλεσμα στην έξοδο για να πραγματοποιηθεί έλεγχος συμπεριφοράς ή της κίνησης των φυσικών αντικειμένων.

Η διαδικασία της κατασκευής και του προγραμματισμού ενός ρομπότ θεωρείται μια ευχάριστη διαδικασία για τους μαθητές, όπου μέσω αυτής έχουν την δυνατότητα να εκπαιδευτούν σε αρκετές έννοιες που αφορούν τις θετικές επιστήμες αλλά και πολλές άλλες. Ένα από τα θετικά σημεία της διαδικασίας είναι ότι οι μαθητές εκπαιδεύονται στην παρατηρητικότητα, στο χειρισμό μιας ρομποτικής συσκευής καθώς και στην κατασκευή της. Μέσω της εκπαιδευτικής ρομποτικής οι μαθητές ζυμώνονται με τον προγραμματισμό, αποκτούν ευχέρεια στην λήψη αποφάσεων και στην επίλυση των



προβλημάτων που έχουν άμεση σχέση με την ρομποτική. Αυτό έχει ως συνέπεια οι μαθητές παίζοντας και λύνοντας πρακτικές ασκήσεις να εκπαιδεύονται στις θετικές επιστήμες με την βοήθεια της εκπαιδευτικής ρομποτικής. Η ευκαιρία για κατασκευή αλλά και η πρακτική εξάσκηση με καινοτόμα εργαλεία, παροτρύνει τους μαθητές να ασχοληθούν περισσότερο με το αντικείμενο και να αποκομίσουν περισσότερες γνώσεις. Ως υπολογιστική σκέψη λογίζεται η δυνατότητα χρησιμοποίησης εννοιών πληροφορικής με σκοπό τον προσδιορισμό και την επίλυση διαφόρων προβλημάτων. Κάνοντας χρήση της εκπαιδευτικής ρομποτικής οι μαθητές βελτιώνουν την ικανότητα τους στην υπολογιστική σκέψη, διότι καλούνται να σχεδιάσουν διάφορους αλγορίθμους, να εντοπίσουν τυχόν σφάλματα, να κάνουν χρήση της επανάληψης και της γενίκευσης και να μάθουν να αποδομούν τα προβλήματά τους. Κατασκευάζοντας και προγραμματίζοντας ρομποτικές κατασκευές οι εκπαιδευόμενοι λύνουν με αλγοριθμικό τρόπο τα προβλήματά τους και αποκτούν μεγαλύτερη εξοικείωση με την τεχνολογία με αποτέλεσμα να αναπτυχθεί η υπολογιστική τους σκέψη. Το ενδιαφέρον των μαθητών γίνεται όλο και μεγαλύτερο για την εκπαιδευτική ρομποτική, προσεγγίζει μεγάλη γκάμα ηλικιών και προκαλεί ενθουσιασμό στα περισσότερα παιδιά. Προσφέρει στους μαθητές ποιοτική μάθηση καθώς οι δραστηριότητες που περιλαμβάνει είναι δημιουργικές και πολύπλευρες.

2.4 Οι προσεγγίσεις της εκπαιδευτικής ρομποτικής

Χρησιμοποιώντας τον όρο εκπαιδευτική ρομποτική πραγματοποιούμε μια διδακτική προσέγγιση ενός εκπαιδευτικού με την μάθηση. Κάνοντας χρήση των ρομπότ ο εκπαιδευτικός πλευρίζει τη γνώση είτε μέσα από τα ρομπότ, είτε από τα ίδια τα ρομπότ. Το μεγάλο πλεονέκτημα της ρομποτικής ως εργαλείο μάθησης είναι ότι οι εκπαιδευόμενοι μαθαίνουν παίζοντας. Αυτό έχει ως αποτέλεσμα μέσω του παιχνιδιού το ενδιαφέρον των μαθητών να μεγαλώνει, να επιστρατεύουν προηγούμενες γνώσεις για την επίλυση του προβλήματος και με την συναναστροφή τους με τα ρομπότ να κατανοούν σε βάθος την λειτουργία τους. Ακολουθώντας τα λεγόμενα του Αλιμήση (2009) σύμφωνα με τον τρόπο που θα χρησιμοποιήσουμε την εκπαιδευτική ρομποτική, θα μπορέσουμε να την χωρίσουμε σε δύο μεγάλες κατηγορίες. Αύτη που η εκπαιδευτική ρομποτική θεωρείται γνωστικό αντικείμενο αλλά και όταν την συναντάμε ως εργαλείο εκμάθησης.



2.4.1 Η εκπαιδευτική ρομποτική ως γνωστικό αντικείμενο

Στην συγκεκριμένη κατηγορία το αντικείμενο μελέτης θεωρείται η ρομποτική ως ένα διαφορετικό αντικείμενο. Στόχος της είναι μέσω των δραστηριοτήτων να δημιουργήσει ένα μαθησιακό περιβάλλον όπου οι εκπαιδευόμενοι, θα παίρνουν μέρος στην λύση των προβλημάτων που αφορούν ρομποτικά ζητήματα όπως η κατασκευή ενός ρομπότ, ο προγραμματισμός του, η τεχνητή του νοημοσύνη και κατά πόσο θα υπήρχε η δυνατότητα όλα αυτά να ενσωματωθούν στο πρόγραμμα σπουδών της πληροφορικής.

2.4.2 Η εκπαιδευτική ρομποτική ως εργαλείο εκπαίδευσης

Στην κατηγορία αυτή γίνεται χρήση της εκπαιδευτικής ρομποτικής ως εκπαιδευτικό εργαλείο, με σκοπό την εκμάθηση κάποιων αντικειμένων όπως τα μαθηματικά, η φυσική και υπάρχει η δυνατότητα να εφαρμοστεί σε κάποιες τάξεις της πρωτοβάθμιας αλλά και σ' όλη την δευτεροβάθμια εκπαίδευση. Ως εκπαιδευτικό εργαλείο η εκπαιδευτική ρομποτική κάνει χρήση πολλών επιστημών (Μαθηματικά, Φυσική, Τεχνολογία, Πληροφορική) κατά συνέπεια καταλήγει να θεωρείται μια διεπιστημονική μαθησιακή δραστηριότητα. Θα μπορούσε πολύ εύκολα σ' ένα πρόγραμμά σπουδών να ενταχθεί ως Project.

Ωστόσο έχοντας αυτές τις δύο προσεγγίσεις της εκπαιδευτικής ρομποτικής είναι δύσκολο να γίνει ο διαχωρισμός τους. Υπάρχει δυνατότητα εμπλοκής της μηχανικής, των μαθηματικών και της φυσικής ακόμη και όταν η εκπαιδευτική ρομποτική λειτουργεί ως ξεχωριστό μαθησιακό αντικείμενο και δίνεται η δυνατότητα κάλυψης και άλλων στόχων που ενδεχομένως να βρίσκονται στο μαθησιακό πρόγραμμα.

2.5 Η εκπαιδευτική ρομποτική σε συνάρτηση με τα STEM

Με την λέξη STEM αναφερόμαστε στις Φυσικές επιστήμες (Science), Τεχνολογία (Technology), Μηχανική (Engineering) και στα Μαθηματικά (Maths). Τα STEM



προσεγγίζουν την εκπαίδευση με έναν ιδιαίτερο τρόπο συνδυάζοντας δύο, ή και περισσότερες επιστήμες με τον φυσικό κόσμο. Η προσέγγιση των STEM προσφέρει στους εκπαιδευόμενους πέραν από τις γνώσεις στον ανάλογο κλάδο, την αντίληψη αλλά και γιατί οι συγκεκριμένες επιστήμες θεωρούνται σημαντικές στην καθημερινή μας ζωή. Η τάση της σημερινής κοινωνίας να ψηφιοποιεί τα περισσότερα μέσα γεννά την ανάγκη, για δημιουργία νέων δεξιοτήτων αλλά και επαγγελμάτων που παλαιότερα δεν υπήρχαν. Έτσι η εκπαιδευτική ρομποτική καλλιεργεί τέτοιες δεξιότητες και κάνει το ενδιαφέρον των εκπαιδευόμενων να αυξάνει για τα STEM. Σύμφωνα με αρκετές έρευνες που έχουν διεξαχθεί στο παρελθόν σε μαθητές όπου πήραν μέρος σε δραστηριότητες εκπαιδευτικής ρομποτικής, βρέθηκε ότι οι συγκεκριμένοι μαθητές βελτίωσαν την ομαδική τους ικανότητα, έγιναν πιο δημιουργικοί και ανέπτυξαν την ικανότητα τους να επιλύουν προβλήματα. Επιπλέον απέσπασαν αρκετές γνώσεις στο κομμάτι του προγραμματισμού. Η εκπαιδευτική ρομποτική θεωρείται μια καινοτόμος μέθοδος η οποία εισάγει ομαλά τους μαθητές στο εκπαιδευτικό μοντέλο της εκπαίδευσης STEM και έχει την δυνατότητα να υλοποιήσει όλους τους σκοπούς που ορίζουν τα STEM με τον εξής τρόπο:

- ✚ **Διεπιστημονικότητα:** γενικότερα η ρομποτική ως επιστήμη περιέχει διάφορες αρχές από άλλους κλάδους όπως η Φυσική, τα Μαθηματικά, η Πληροφορική και η Μηχανική. Χρησιμοποιώντας την συνδυαστική διδασκαλία οι μαθητές έχουν την δυνατότητα ν' αναπτύξουν τις γνώσεις τους σ' αυτούς του κλάδους.
- ✚ **Επίλυση προβλημάτων:** πραγματοποιώντας οι μαθητές μια εκπαιδευτική ρομποτική κατασκευή θα βρεθούν αντιμέτωποι με αρκετά προβλήματα τα οποία θα πρέπει να επιλύσουν. Με τον τρόπο αυτό η μάθηση διαφοροποιείται από την θεωρητική προσέγγιση επίλυσης ενός προβλήματος, καθώς οι μαθητές είναι αναγκασμένοι να ασχοληθούν πρακτικά και όχι μόνο με το πρόβλημα.
- ✚ **Δημιουργικότητα:** η δημιουργικότητα και οι ιδέες των μαθητών ανθίζουν κατά την επίλυση ενός προβλήματος κατασκευαστικού ή προγραμματιστικού χαρακτήρα.



2.6 Συμπεράσματα

Ως κατακλείδα, και λαμβάνοντας υπόψη τις ως άνω παρατηρήσεις, σημειώνεται ότι με την εκπαιδευτική ρομποτική εισάγεται δυναμικά ένα καινούργιο μέσο απόκτησης ψηφιακών δεξιοτήτων, το οποίο είναι ικανό να προσελκύσει με μεγάλη επιτυχία το ενδιαφέρον των παιδιών. Με άλλα λόγια, οι μαθητές, μέσω αυτής της μεθόδου πορεύονται προς την κατάκτηση της γνώσης με διασκεδαστικό τρόπο, νομίζοντας πως συμμετέχουν σε ένα εξελισσόμενο παιχνίδι, ενισχύοντας έτσι παράλληλα τη δημιουργικότητα, όπως επίσης και την ομαδικότητά τους. Θα μπορούσε δε να θεωρηθεί ότι υφίστανται δύο διαφορετικές προσεγγίσεις της Εκπαιδευτικής Ρομποτικής: Πρώτον, αυτή της αντιμετώπισής της ως διακριτού γνωστικού αντικείμενου που μπορεί να αντιμετωπιστεί ως ξεχωριστό αντικείμενο μελέτης το οποίο μπορεί να χρησιμοποιηθεί στα πλαίσια μιας διαθεματικής προσέγγισης των σχολικών εργασιών ή σε εξωσχολικές δραστηριότητες, και δεύτερον αυτή της αντιμετώπισής της ως επικουρικού της εκπαιδευτικής διαδικασίας εργαλείου, με το οποίο διευκολύνεται η διδασκαλία σχεδόν όλων των μαθημάτων του σχολικού προγράμματος. Επιπρόσθετα, η εκπαιδευτική ρομποτική ιδωμένη από τη σκοπιά του STEM φαίνεται να εισάγει μια ριζοσπαστική διδακτική προσέγγιση που ενεργοποιεί και εμπλέκει πολλές έννοιες: Επιστήμη, Μαθηματικά, Μηχανική, Τεχνολογία, κ.α.

3. Ο μικροελεγκτής Arduino

3.1 Ιστορικά στοιχεία του Arduino

Την χρονιά του 2005 υπήρξε ένα εγχείρημα από μαθητές με σκοπό να δημιουργήσουν μια συσκευή για έλεγχο διαδραστικών σχεδίων, όπου το κόστος της δεν θα ήταν απαγορευτικό και θα είχαν πρόσβαση αρκετά άτομα. Λαμβάνοντας υπόψιν τους τα παραπάνω ο Massimo Banzi και ο David Cueartielles οι οποίοι θεωρούνται ιδρυτές του Arduino, ξεκίνησαν μια παραγωγή πλακετών σε ένα εργοστάσιο της Ιβρέας μιας πόλης της Βόρειας Ιταλίας κοντά στο Τορίνο. Εμπνεύστηκαν το όνομα από τον



Αρντουίνο της Ιβρέας ο οποίος ήταν βασιλιάς της Λομβαρδίας το διάστημα 1002-1014. Το Arduino βασίστηκε σε μια διακλάδωση της πλατφόρμας Wiring και θεωρείται λογισμικό ανοιχτού κώδικα. Η γλώσσα προγραμματισμού του μοιάζει αρκετά με την C++ με αρκετές απλοποιήσεις και αλλαγές και ανήκει στο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

3.2 Οδηγός εκμάθησης Arduino

3.2.1 Τα περιβάλλοντα προγραμματισμού του Arduino και ο μικροελεγκτής

Μιλώντας για το Arduino αναφερόμαστε σε μια ηλεκτρονική πλατφόρμα η οποία έχει ως χαρακτηριστικό τον ανοιχτό κώδικα, τον σχεδιασμό και επιπλέον παρέχει στον χρήστη οικείο υλικό αλλά και εύκολο λογισμικό. Η δημιουργία του βασίστηκε στις ανάγκες καλλιτεχνών, σχεδιαστών και γενικότερα σ' οποιοδήποτε του αρέσει να πραγματοποιεί το χόμπι του ή και την δραστηριότητα του δημιουργώντας αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα.

Προσεγγίζοντας λίγο περισσότερο το τεχνικό κομμάτι του Arduino είναι καλό να αναφέρουμε ότι περιέχει ένα κύκλωμα το οποίο κάνει χρήση ενός μικροελεγκτή. Ο οποίος έχοντας έναν αριθμό από πύλες τις προσαρμόζει ανάλογα με τις ανάγκες του χρήστη σαν εισόδους και εξόδους στο συγκεκριμένο κύκλωμα. Η διαχείριση αυτών των εισόδων και των εξόδων πραγματοποιείται με προγραμματισμό του Arduino IDE και η γλώσσα μοιάζει αρκετά με την C/C++. Μέσω της ιστοσελίδας του Arduino (<http://arduino.cc/>) μας παρέχεται αρκετό υλικό καθώς και χρήσιμες πληροφορίες, όπως επίσης έχουμε την ευκαιρία να κατεβάσουμε το περιβάλλον προγραμματισμού από την συγκεκριμένη σελίδα (<http://arduino.cc/en/Main/Software>).

Πέρα από την κύρια έκδοση του περιβάλλοντος του Arduino IDE, υπάρχουν κι άλλες εκδόσεις παραλλαγμένες. Μια τέτοια έκδοση είναι αυτή του scratch που μας δίνεται η δυνατότητα να υλοποιήσουμε προγράμματα για το Arduino. Ονομάζεται "Scratch for Arduino" και μεγάλο πλεονέκτημα του είναι ο οπτικός προγραμματισμός ανοιχτού κώδικα, ο οποίος είναι αρκετά οικείος στα παιδιά. Παρέχεται δωρεάν και εξυπηρετεί πολύ εκπαιδευτικούς σκοπούς σε σχέση με την απλή συγγραφή κώδικα ο



οποίος ενδείκνυται για μεγαλύτερες τάξεις. Τέλος υπάρχουν και αρκετά web περιβάλλοντα με οπτικές εκδόσεις όπως το “BlocklyDuino” και το “ArduinoMio” τις οποίες μπορούμε να χρησιμοποιήσουμε κατευθείαν από την ιστοσελίδα τους. Στα παρακάτω κεφάλαια θα προσεγγίσουμε πως λειτουργεί ο μικροελεγκτής του Arduino και πως προγραμματίζεται με την βοήθεια της πλατφόρμας του Arduino IDE. Θα πραγματοποιηθεί ένας αναλυτικός οδηγός χρήσης και θα δοθούν αρκετά παραδείγματα για την εξοικείωση του χρήστη. Ο οδηγός θα ολοκληρώνεται με εκπαιδευτικά σενάρια με την βοήθεια του Arduino.

3.2.2 Απαραίτητα υλικά για την χρήση του Arduino

Στο Arduino το κύκλωμα που υπάρχει θεωρείται ανοικτό, κατά συνέπεια μας δίνεται η δυνατότητα να γνωρίζουμε από τον κατασκευαστή του τα μέρη του και τον σχεδιασμό του. Έτσι, το υλικό που προέρχεται από τους δημιουργούς του Arduino από την Ιταλία παρέχεται στο κοινό δωρεάν, προσφέροντας διάφορα συμβατά προγράμματα που μπορούν να χρησιμοποιηθούν από το οποιοδήποτε. Ο μόνος περιορισμός που έθεσαν από την αρχή ήταν το όνομα που είχαν επιλέξει να παραμείνει σ’ αυτούς και οποιοσδήποτε τρίτος δώσει όνομα σε κάτι σχετικό να χρησιμοποιήσει διαφορετικό όνομα. Οι δημιουργοί δέχτηκαν και συμφώνησαν ότι θα υπάρξουν διάφορες εκδόσεις που οποιοδήποτε όνομα και να έδιναν οι τρίτοι η κατάληξη τους στο τέλος θα περιείχε το -ino. Το υλικό διατίθεται στην επίσημη ιστοσελίδα από τους κατασκευαστές (<http://store.arduino.cc/>) ενώ αρκετοί είναι αυτοί που τα προωθούν στην Ελλάδα (<http://arduino.cc/en/Main/Buy>).

Τα εξαρτήματα που θα χρειαστούμε αρχικά είναι τα εξής:

- ✚ Μια πλακέτα Arduino uno R3 την οποία θα κάνουμε χρήση στα παρακάτω παραδείγματα. Η συγκεκριμένη πλακέτα περιέχει 14 ψηφιακές εισόδους/εξόδους και 6 αναλογικές εισόδους. Για τις πρώτες μας εφαρμογές ο όγκος των εισόδων και των εξόδων είναι ικανοποιητικός, και η σύνδεση θα πραγματοποιηθεί με το αντίστοιχο καλώδιο USB.
- ✚ Τα καλώδια για την σύνδεση των εισόδων και των εξόδων.
- ✚ Ένα breadboard στο οποίο θα υλοποιούμε όλα τα κυκλώματα που θέλουμε να ελέγξουμε με το Arduino.



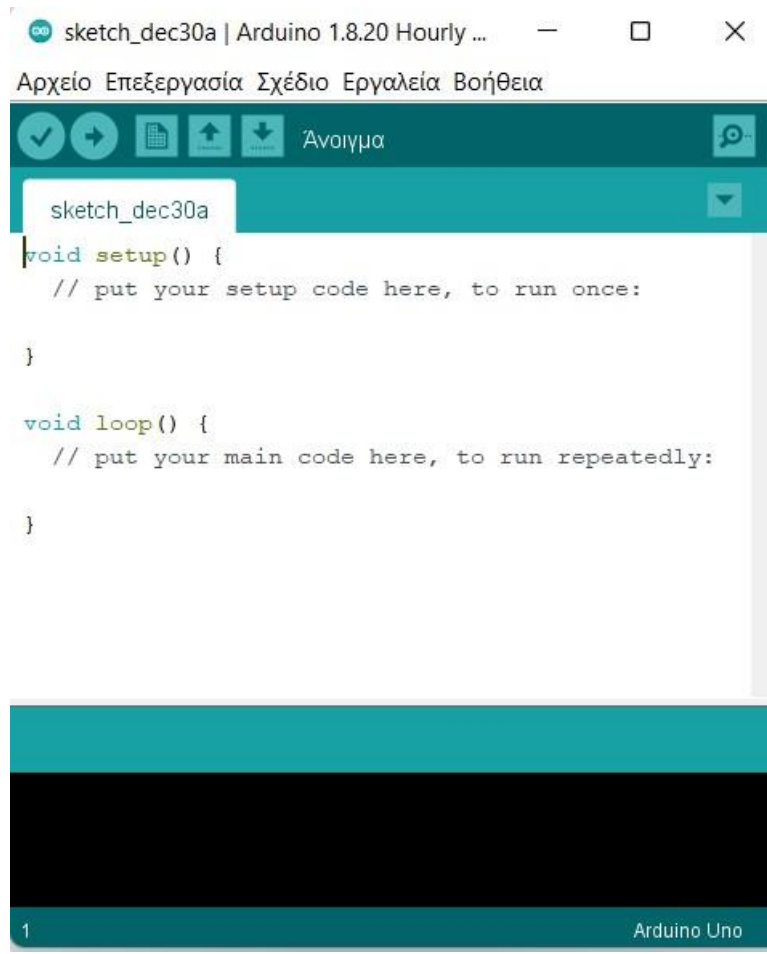
- ✚ Τα αρχικά εξαρτήματα τα οποία χρησιμοποιεί ένας αρχάριος για την υλοποίηση των κυκλωμάτων. (Led, αντιστάσεις, πυκνωτές, ποντεσιόμετρα, φωτοαντιστάσεις, κουμπιά)

Επίσης θα προμηθευτούμε ένα κιτ αρχαρίων που περιέχει όλα τα απαραίτητα υλικά για έναν αρχάριο (Breadboard, usb, led, αντιστάσεις, πυκνωτές, φωτοαντιστάσεις, ποντεσιόμετρα, κουμπιά, καλώδια διασύνδεσης). Ένα τέτοιο κιτ είναι το Arduino starter kit που έχουμε την δυνατότητα να το προμηθευτούμε από αρκετά καταστήματα (<https://www.skroutz.gr/s/20597293/Uno-R3-Learning-Kit.html>).

3.2.3 Εγκατάσταση του περιβάλλοντος Arduino IDE

Στο Arduino πέρα από τις συνδεσμολογίες και τα διάφορα κυκλώματα για να μπορέσει να γίνει έλεγχος της κατασκευής θα πρέπει να υλοποιηθεί προγραμματισμός. Κατά συνέπεια θα πρέπει να γίνει η εγκατάσταση του περιβάλλοντος προγραμματισμού του Arduino IDE. Ο κώδικας που θα χρησιμοποιήσουμε μοιάζει αρκετά με την γλώσσα προγραμματισμού C/C++ ο οποίος είναι αναγκαίο να μεταγλωττιστεί και να φορτωθεί στην μονάδα μας. Το συγκεκριμένο περιβάλλον είναι σχεδιασμένο για τα Windows ωστόσο λειτουργεί και σε άλλα λογισμικά όπως τα Linux και το Mac, η λήψη του είναι εντελώς δωρεάν και μπορούμε να το κατεβάσουμε από την επίσημη ιστοσελίδα.

(<http://arduino.cc/en/Main/Software>)



Εικόνα 1-Περιβάλλον προγραμματισμού Arduino IDE

Το παραπάνω περιβάλλον προγραμματισμού παρέχεται και στα ελληνικά και έχει πολλά παραδείγματα χρήσης για τους αρχάριους. Το μενού είναι αρκετά απλό και ο προγραμματισμός μπορεί να υλοποιηθεί εδώ. Εμείς θα το χρησιμοποιήσουμε και αυτό αλλά και το περιβάλλον προγραμματισμού στο Tinkercad, το οποίο είναι σχεδόν ίδιο με την μόνη διαφορά ότι όλες οι λειτουργίες του Arduino γίνονται εικονικά.

3.2.4 Το ρεύμα λειτουργίας του Arduino

Στο Arduino υπάρχει η επιλογή να δουλέψει η συσκευή με την θύρα Usb που παρέχεται από τον υπολογιστή μας, αλλά και με ξεχωριστή παροχή ρεύματος την οποία έχουμε την δυνατότητα να πάρουμε από την μπαταρία. Η πλακέτα μας παρέχει σταθερή τάση της τάξεως των 5V στις εξόδους.

Επιπλέον μας δίνεται η εναλλακτική να χρησιμοποιήσουμε εξωτερική πηγή για την παροχή του ρεύματος από εξωτερικό βύσμα, η τάση λειτουργίας που προτείνετε είναι



στα 7 – 12 V για να λειτουργεί κανονικά η πλακέτα και να δίνει σταθερά 5V κατά την έξοδο του. Η σύνδεση της μπαταρίας μπορεί να πραγματοποιηθεί στις κατάλληλες εξόδους της πλακέτας (VCC IN(+) και GND(-)). Εναλλακτικά αν η συσκευή είναι συνδεδεμένη μόνιμα με την θύρα Usb τότε λειτουργεί στα 5V χωρίς κάποιο ιδιαίτερο πρόβλημα.

3.2.5 Τα Pins του Arduino (Είσοδοι/Εξοδοι)

Χρησιμοποιώντας το Arduino Uno R3 θα συναντήσουμε 14 ψηφιακές θύρες εισόδου και εξόδου ανάλογα με το πως θα δηλωθούν στον προγραμματισμό, και 6 ξεχωριστές αναλογικές θύρες εισόδου. Η ονοματολογία των ψηφιακών θυρών ξεκινάει από τον αριθμό 0 και φτάνει μέχρι τον αριθμό 13. Αντιθέτως οι 6 αναλογικές σε κάθε θύρα υπάρχει το γράμμα A(Analog) και από δίπλα ο αντίστοιχος αριθμός όπου ξεκινάει από το 0 και καταλήγει έως και το 5 (A0-A5). Η τάση στην έξοδο από τα pins είναι από 0V μέχρι 5V. Στις 14 ψηφιακές θύρες και συγκεκριμένα στους αριθμούς 3, 5, 6, 9, 10, 11 υπάρχει η δυνατότητα να τις χρησιμοποιήσουμε μέσω του προγραμματισμού ως ψευδοαναλογικές εξόδους. Τις ονομάζουμε PWM (Pulse width modulation) και έχουν την δυνατότητα να αναπαριστήσουν μια αναλογική έξοδο.

Συμπερασματικά λοιπόν για εισόδους και εξόδους στο Arduino μας παρέχονται:

- ✚ **Ψηφιακή είσοδο:** Κάνουμε χρήση των 14 ψηφιακών εισόδων από την θύρα 0 έως και την 13. Χρησιμοποιώντας τις θύρες ψηφιακά ως είσοδο έχουμε την ευκαιρία να δίνουμε την τάση των 0V ή 5V γράφοντας κώδικα. Το LOW ανταποκρίνεται στα 0V ενώ το HIGH στα 5V. Στη συνέχεια του οδηγού θα γίνει ανάλυση κάποιων βασικών εντολών.
- ✚ **Ψηφιακή έξοδο:** Κάνουμε χρήση των 14 ψηφιακών εξόδων από την θύρα 0 έως και την 13. Χρησιμοποιώντας τις θύρες ψηφιακά ως έξοδο έχουμε την ευκαιρία να δίνουμε την τάση των 0V ή 5V γράφοντας κώδικα. Το LOW ανταποκρίνεται στα 0V ενώ το HIGH στα 5V. Στη συνέχεια του οδηγού θα γίνει ανάλυση κάποιων βασικών εντολών.
- ✚ **Αναλογική είσοδο:** Στις αναλογικές εισόδους από A0-A5 μας παρουσιάζεται η δυνατότητα να παίρνουμε τιμές του ρεύματος από 0V έως 5V.
- ✚ **Αναλογική έξοδο:** Στις αναλογικές εξόδους έχουμε την δυνατότητα να κάνουμε χρήση στις 6 ψηφιακές θύρες PWM. Δηλαδή σε όσες θύρες δίπλα



απ' τον αριθμό έχουν την – (3, 5, 6, 9, 10, 11). Το ρεύμα εξόδου το οποίο θα λάβουμε είναι πάλι μεταξύ των 0V και 5V.

Είναι αναγκαίο να τονίσουμε ότι οι θύρες για να χρησιμοποιηθούν θα πρέπει να αρχικοποιηθούν. Αυτό συμβαίνει με την εντολή `pinMode()` η οποία δηλώνει την θύρα που θα επιλέξουμε ως τι θέλουμε να την χρησιμοποιήσουμε. Η συγκεκριμένη εντολή όπως και άλλες συναρτήσεις και εντολές θα εξηγηθούν παρακάτω.

Τις θύρες 0 και 1 συνήθως τις χρησιμοποιούμε για την παρακολούθηση της επικοινωνίας με τον υπολογιστή. Οπότε καλό θα ήταν να μην τις χρησιμοποιούμε παρά μόνο όταν δεν έχουμε άλλη θέση στις θύρες.

Τέλος στο Arduino Uno R3 που θα ασχοληθούμε εμείς η θύρα 13 περιέχει ήδη ένα led το οποίο μπορούμε να το χρησιμοποιήσουμε για σχετικά πρότζεκτ ή λειτουργίες.

3.2.6 Βασικές λειτουργίες του Arduino Uno R3 και προγραμματισμός

Αφού ολοκληρώσουμε την εγκατάσταση του Arduino IDE έχουμε την δυνατότητα να ξεκινήσουμε να γράφουμε κώδικα. Αυτό μπορούμε να το κάνουμε και στην πλατφόρμα του Tinkercad ωστόσο οποιαδήποτε ενέργεια κάνουμε με το Arduino μας θα πραγματοποιηθεί εικονικά.

Το Arduino χρησιμοποιεί δύο βασικές συναρτήσεις την `setup()` και την `loop()` που είναι αρκετά απλές και κάνουν την εξής δουλειά:

🚦 **Setup ()**: Στην συγκεκριμένη συνάρτηση όσες εντολές βάζουμε ενδιάμεσα από τις αγκύλες της συνάρτησης υλοποιούνται μόνο μία φορά, όταν πραγματοποιηθεί δηλαδή η εκκίνηση της μονάδας μας. Συχνό φαινόμενο είναι στην εν λόγω συνάρτηση να συναντούμε εντολές που αφορούν αρχικοποίηση μεταβλητών, αλλά και τον χαρακτηρισμό των εισόδων και των εξόδων. (Τον τρόπο δηλαδή που θα χρησιμοποιήσουμε το `pin`).

🚦 **Loop ()**: Στην συγκεκριμένη συνάρτηση γράφουμε την βασική δομή του προγράμματος μας. Όσες εντολές εμπεριέχονται ενδιάμεσα από τις αγκύλες της `loop()` θα πραγματοποιηθούν ακολουθιακά και με το πέρας των εντολών θα επαναληφθούν από την αρχή, μέχρις ότου το Arduino σταματήσει να έχει ρεύμα ή αν πατηθεί το πλήκτρο `reset`.



Οπότε το κύριο χαρακτηριστικό του Arduino είναι ότι αρχικά λειτουργεί η συνάρτηση Setup () μία φορά και έπειτα ακολουθεί η Loop () συνεχώς, μέχρις ότου να τερματίσουμε την συσκευή ή να πιέσουμε το πλήκτρο Reset.

Εφόσον επαναλάβουμε την διαδικασία και πατήσουμε ξανά το Reset τρέχει η συνάρτηση Setup (), εν συνεχεία ακολουθεί η Loop () ξανά και ξανά με τον ίδιο τρόπο που βάζουμε στο ρεύμα τον μικροελεγκτή την πρώτη φορά. Αν πραγματοποιήσουμε αλλαγές στον κώδικα μας και τις φορτώσουμε στο Arduino, πατώντας το Reset γίνονται οι αλλαγές.

Παρακάτω ακολουθεί παράδειγμα δομής του προγράμματος του Arduino:

```
Void setup() {
```

```
// Οι συγκεκριμένες εντολές θα τρέξουν κατά την εκκίνηση της συσκευής ή  
πιέζοντας το κουμπί του Reset//
```

```
}
```

```
Void loop() {
```

```
// Οι συγκεκριμένες εντολές θα τρέξουν ακολουθιακά συνεχόμενα, μέχρι να  
απενεργοποιηθεί το Arduino ή να πατηθεί το πλήκτρο Reset. //
```

```
}
```

Δήλωση των μεταβλητών

Στις περισσότερες γλώσσες προγραμματισμού έχουμε την δυνατότητα να κάνουμε δήλωση μεταβλητών. Το ίδιο ισχύει και για το Arduino το οποίο και αυτό παρέχει αρκετούς τύπους μεταβλητών. Παρακάτω θα παρουσιάσουμε κάποιους τύπου μεταβλητών οι οποίοι θα είναι οικείοι στους αρχάριους χρήστες :

- ✚ **Boolean** : Παρομοιάζουμε με το όνομα Boolean τις μεταβλητές που παίρνουν τιμή 0 ή 1 (True ή False).
- ✚ **Int**: Παρομοιάζουμε με το όνομα Int τις μεταβλητές οι οποίες παίρνουν ακέραια τιμή από -32768 έως 32767.
- ✚ **Float**: Παρομοιάζουμε με το όνομα Float όλες τις μεταβλητές που παίρνουν δεκαδικούς αριθμούς.
- ✚ **Byte**: Παρομοιάζουμε με το όνομα Byte τις μεταβλητές που παίρνουν τιμές από 0 έως 255.



- ✚ **Long:** Παρομοιάζουμε με το όνομα Long τις μεταβλητές που είναι ακέραιες και παίρνουν τιμές από -2147483648 έως 2147483647.
- ✚ **Char:** Παρομοιάζουμε με το όνομα Char τις μεταβλητές που παίρνουν έναν χαρακτήρα (Byte).
- ✚ **String:** Παρομοιάζουμε με το όνομα String τις μεταβλητές οι οποίες μπαίνουν σε ένα πίνακα χαρακτήρων.

Θα ακολουθήσει παράδειγμα δήλωσης μεταβλητών:

```
Int ledPin = 7; // Πραγματοποιώ τον ορισμό του αριθμού του pin για το Led το οποίο θέλω.  
Float SinVal; // Πραγματοποιώ τον ορισμό μια πραγματικής μεταβλητής με το όνομα SinVal.
```

Σχόλια στο Arduino

Στις περισσότερες γλώσσες προγραμματισμού έχουμε την δυνατότητα να προσθέσουμε σχόλια, με σκοπό αν κάποιος χρειάζεται επεξήγηση σε κάποιο σημείο του κώδικα. Το ίδιο συμβαίνει και στο Arduino όπου για να προσθέσω σχόλια θα πρέπει πριν γράψω αυτό που θέλω να βάλω δύο κάθετες γραμμές // και το υπόλοιπο κείμενο που θα γράψω απλά αγνοείται από τον υπολογιστή. Αν θέλω το σχόλιο που θα κάνω να βρίσκεται σε παραπάνω από μία γραμμή θα πρέπει να βάλω τα εξής σύμβολα /* */. Θα ακολουθήσει παράδειγμα για την χρήση και των δύο τρόπων.

Παράδειγμα σχολίων στο Arduino:

```
Int ledPin= 7; // Αναθέτω τον αριθμό του Arduino που θα πάρει το Led  
/* Στον παρακάτω κώδικα θα γίνει προγραμματισμός ενός Led το οποίο θα  
Κάνει Blink για 1 δευτερόλεπτο. */
```

Συνάρτηση διαχείρισης των θυρών εισόδου και εξόδου στο Arduino

Παρατηρώντας τον μικροελεγκτή του Arduino θα δούμε ότι η βασική του λειτουργία στηρίζεται στον έλεγχο των εισόδων και των εξόδων. Ανάλογα τον κώδικα που θα γράψουμε άλλοτε ο μικροελεγκτής δίνει ρεύμα και άλλοτε το παίρνει. Αυτό που είναι απαιτούμενο να κάνουμε εμείς για να επιτύχουμε την συγκεκριμένη διεργασία είναι να χαρακτηρίσουμε την κάθε είσοδο ή την κάθε έξοδο του μικροελεγκτή. Κατά συνέπεια κάνοντας χρήση την συνάρτηση pinMode (pin, Mode) στο πρώτο μέρος την παρένθεσης εκεί που λέει pin βάζουμε τον αριθμό της θύρας. Και στο δεύτερο μέρος



της παρένθεσης εκεί που έχει το Mode την χαρακτηρίζουμε είτε ως είσοδο είτε ως έξοδο. Οι λέξεις που χρησιμοποιούμε για τον χαρακτηρισμό είναι τον INPUT αν θέλουμε την θύρα να την χρησιμοποιήσουμε ως είσοδο, και αντίστοιχα αν θέλουμε να την χρησιμοποιήσουμε ως έξοδο χρησιμοποιούμε την λέξη OUTPUT. Έχουμε την δυνατότητα επιλογής από τα 14 ψηφιακά pins τα οποία είχαμε αναφέρει, όπου τα έξι λογίζονται ως PWM αλλά και από τις 6 αναλογικές θύρες.

Παράδειγμα:

```
pinMode (7, OUTPUT);  
pinMode (ledpin, OUTPUT);  
pinMode (A0, INPUT);
```

Ψηφιακή είσοδο και έξοδο στο Arduino

Τα 14 pins που υπάρχουν στο Arduino υπάρχει η δυνατότητα να λειτουργούν είτε ως ψηφιακές εισόδους είτε ως ψηφιακές εξόδους. Δηλαδή όταν το pin χρησιμοποιείται ως έξοδο μπορεί να δώσει από 0V έως 5V. Αυτό συμβαίνει όταν χρησιμοποιήσουμε την εντολή **digitalWrite(pin, Value)**. Όπου το pin στο πρώτο μέρος την παρένθεσης αντιπροσωπεύει την θύρα και το value στο δεύτερο μέρος την τάση την οποία μπορούμε να δώσουμε. Οι τιμές τις τάσεις που μπορούμε να χρησιμοποιήσουμε είναι προκαθορισμένες. Οπότε χρησιμοποιούμε την λέξη LOW αν θέλουμε η τάση μας να έχει 0V και αν θέλουμε η τάση μας να έχει την τιμή των 5V χρησιμοποιούμε την λέξη HIGH.

✚ LOW: Μας δίνει 0V κατά την έξοδο

✚ HIGH: Μας δίνει 5V κατά την έξοδο

Όταν θέλουμε να χρησιμοποιήσουμε ένα από τα pins μας ως είσοδο τότε χρησιμοποιούμε την εντολή **digitalRead(Pin)**. Η είσοδος έχει την δυνατότητα να διαβάσει μια τιμή από 0V έως 5V. Το Pin μέσα στην παρένθεση αναφέρεται στο νούμερο της εισόδου που θα χρησιμοποιήσουμε από το Arduino, ενώ η συνάρτηση κάνει επιστροφή του ονόματος για την τιμή της εισόδου. Οι τιμές για την τάση της εισόδου κυμαίνονται το ίδιο όπως και στην έξοδο, δηλαδή από 0V έως 5V και χαρακτηρίζονται ακριβώς με τις ίδιες λέξεις LOW και HIGH.

✚ LOW: Μας δίνει 0V κατά την είσοδο.

✚ HIGH: Μας δίνει 5V κατά την έξοδο.



Προσοχή: Η κάθε θύρα πρέπει να έχει οριστεί είτε ως είσοδο είτε ως έξοδο στην συνάρτηση `Void Setup()` με την συνάρτηση `pinMode` αλλιώς δεν θα λειτουργήσει όπως εμείς θέλουμε.

Παράδειγμα εξόδου:

```
digitalWrite (ledpin, HIGH);
```

Παράδειγμα αρχικοποίησης εξόδου:

```
pinMode (7, OUTPUT);
```

Παράδειγμα εισόδου:

```
Val = digitalRead(ledpin);
```

Παράδειγμα αρχικοποίησης εισόδου:

```
pinMode (7, INPUT)
```

Αναλογική είσοδο και έξοδο στο Arduino

Ως αναλογικές εξόδους στο Arduino έχουμε την δυνατότητα να χρησιμοποιήσουμε τα PWM, τα οποία θεωρούνται έξοδοι παλμοκωδικής διαμόρφωσης. Τα Pin που μπορούμε να χρησιμοποιήσουμε είναι τα (3, 5, 6, 9, 10, 11) όποια δηλαδή έχουν την παύλα από δίπλα. Οπότε κάνοντας χρήση της αναλογικής εξόδου οι τιμές που θα πάρουμε είναι από 0 μέχρι το 255 σε ένα διάστημα 0V έως 5V. Αυτό μπορούμε να το επιτύχουμε συντάσσοντας την εντολή **analogWrite (pin, Value)**. Όπου pin τοποθετείται ο αριθμός της θύρας του Arduino και value τοποθετείται η τάση η οποία είναι από 0V έως 5V. Επειδή η εντολή που θα χρησιμοποιήσουμε αναφέρεται σε αναλογικά όργανα υπάρχει η περίπτωση να δώσουμε και ενδιάμεσες τιμές, θα ακολουθήσει παράδειγμα όπου θα γίνει πιο κατανοητή η λειτουργία. Να υπενθυμίσουμε βέβαια πως για αναλογική έξοδο μπορώ να χρησιμοποιήσω μόνο τα PWM, οποιαδήποτε άλλη θύρα από τις ψηφιακές θύρες δεν θα λειτουργήσει.

Παράδειγμα αναλογικής εξόδου:

```
analogWrite (7, 122);
```

Διπλωματική Εργασία



```
/* Το 122 αντιστοιχεί στα 2.5V και υπάρχει η δυνατότητα να βάλουμε τιμές από το  
0 μέχρι το 255 όταν χρησιμοποιούμε αναλογικές εξόδους */
```

Παράδειγμα αρχικοποίησης αναλογικής εξόδου:

```
pinMode (7, OUTPUT);
```

Στις αναλογικές εισόδους τα pin που μπορούμε να κάνουμε χρήση είναι διαφορετικά και ονοματίζονται από το A0 έως το A5. Υπάρχει η επιλογή να κάνουμε χρήση ενός αναλογικού οργάνου (π.χ ποντεσιόμετρο) και να το δηλώσουμε ως είσοδο. Η συνάρτηση που θα πρέπει να χρησιμοποιήσουμε είναι η **analogRead(pin)**, όπου pin είναι ο αριθμός της θύρας του Arduino, ενώ η συγκεκριμένη συνάρτηση κάνει επιστροφή με το όνομα της την τιμή εισόδου. Κατά συνέπεια κάνουμε χρήση μίας μεταβλητής για την καταχώρηση της συγκεκριμένης τιμής. Έτσι λοιπόν η τιμή εισόδου παίρνει τιμές από 0 έως και 1023.

Παράδειγμα αναλογικής εισόδου:

```
Int r = analogRead(A3);  
// Το int δηλώνει πως η μεταβλητή από μπροστά είναι ακέραια.
```

Παράδειγμα αρχικοποίησης αναλογικής εισόδου:

```
pinMode (A3, INPUT);
```

Η συνάρτηση καθυστέρησης delay()

Ένα θέμα το οποίο έχουμε να διαχειριστούμε στο Arduino είναι η καταγραφή του χρόνου αλλά και η διαχείριση του. Υπάρχουν πολλές συναρτήσεις τις οποίες μπορούμε να συντάξουμε γι' αυτό το σκοπό. Με την συνάρτηση delay() έχουμε την ευκαιρία να ορίσουμε μια καθυστέρηση στο πρόγραμμα για όσο εμείς θέλουμε.

Συντάσσετε με το εξής τρόπο: **delay(time)**



Και όπου time μπαίνει ο χρόνος που θέλουμε να κάνουμε καθυστέρηση στο πρόγραμμα σε ms (1/1000 sec).

Παράδειγμα delay():

```
Delay(1000); //Πραγματοποιεί καθυστέρηση του προγράμματος για 1000 ms=1 sec  
Delay(300); // Πραγματοποιεί καθυστέρηση του προγράμματος για 300 ms=0.3 sec
```

Η συνάρτηση καθυστέρησης delayMicroseconds()

Η συνάρτηση καθυστέρησης delayMicroseconds() την χρησιμοποιούμε με το ίδιο τρόπο που χρησιμοποιούμε την delay(), με την μόνη διαφορά πως ο χρόνος καθυστέρησης πλέον απεικονίζεται σε microseconds (1/10⁶ sec).

Παράδειγμα delayMicroseconds() :

```
delayMicroseconds(10); /* Θα σταματήσει την εκτέλεση του προγράμματος στο  
σημείο για 10μsec = 1/100000 sec */
```

33

Η συνάρτηση της καταγραφής του χρόνου millis()

Ο μικροελεγκτής του Arduino περιέχει ένα ρολόι το οποίο είναι ενσωματωμένο. Κάθε φορά που ενεργοποιείται η συσκευή ξεκινάει ταυτόχρονα και το ρολόι και σταματάει όταν απενεργοποιηθεί η συσκευή ή όταν πατήσουμε το κουμπί του Reset.

Συντάσσοντας την συνάρτηση millis() το Arduino μας κάνει επιστροφή το χρόνο σε milliseconds από την στιγμή που έχει ξεκινήσει ο μικροελεγκτής μας. Με τον τρόπο αυτό η συσκευή μας παρέχει βοήθεια να κάνουμε καταμέτρηση του χρόνου στα προγράμματα που υλοποιούμε, αποφεύγοντας τον κίνδυνο να ξεχάσουμε κάτι που χρειαζόμαστε. Αν χρειαστούμε έλεγχο σε ένα διάστημα που κάτι έχει συμβεί (π.χ το πότε έχει πατηθεί τελευταία φορά ένα κουμπί), έχουμε την δυνατότητα να πραγματοποιήσουμε καταγραφή του γεγονότος σε μια μεταβλητή χρόνου και να κάνουμε την αφαίρεση του χρόνου από τον επόμενο.

Παράδειγμα millis() :

Διπλωματική Εργασία



```
lastPressButton = millis();  
if (lastPressButton - millis() > 1000) /* Εδώ πραγματοποιείται η αφαίρεση που  
αναφερθήκαμε πάνω. Το if θα εξηγήσουμε πως λειτουργεί λίγο παρακάτω */
```

Η συνάρτηση αντιστοίχισης τιμών map()

Στον μικροελεγκτή Arduino είναι αρκετές οι περιπτώσεις που είναι αναγκαίο να πραγματοποιήσουμε αντιστοίχιση τιμών ενός πεδίου με ένα άλλο. Η συγκεκριμένη μαθηματική πράξη είναι αρκετά απλοϊκή, ωστόσο η συσκευή μας δίνει την ευκαιρία να το υλοποιήσουμε με την συνάρτηση map() η οποία λειτουργεί με τον εξής τρόπο:

```
Map( <τιμή>, <Κατώτατο_όριοA>, <Ανώτατο_όριοA>,  
<Κατώτατο_όριοB>, <Ανώτατο_όριοB>)
```

Επεξήγηση της map():

- ✚ Τιμή: Εδώ τοποθετείται η τιμή που θέλουμε να κάνουμε μετατροπή.
- ✚ Κατώτατο όριοA: Εδώ τοποθετείται το κατώτατο όριο του διαστήματος της αρχικής τιμής.
- ✚ Ανώτατο όριοA: Εδώ τοποθετείται το ανώτατο όριο του διαστήματος της αρχικής τιμής.
- ✚ Κατώτατο όριοB: Εδώ τοποθετείται το κατώτατο όριο του διαστήματος της τελικής τιμής (Αυτό που θέλω να κάνω μετατροπή).
- ✚ Ανώτατο όριοB: Εδώ τοποθετείται το ανώτατο όριο του διαστήματος της τελικής τιμής (Αυτό που θέλω να κάνω μετατροπή).

Παράδειγμα της map():

```
Val = map(val, 0, 1023, 0, 172); /* Πραγματοποιώ μια μετατροπή σε μία αναλογική  
τιμή που πήρα από ένα ποτενσιόμετρο (0 έως 1023) σε μία τιμή για έναν  
σερβοκινητήρα */
```



Η σειριακή θύρα επικοινωνίας του Arduino

Η επικοινωνία του Arduino με τον Η/Υ πραγματοποιείται μέσω μιας σειριακής θύρας επικοινωνίας. Για την σύνδεση του Arduino με τον Η/Υ χρησιμοποιείται ένα καλώδιο USB, ενώ όταν θέλουμε να υλοποιήσουμε μια πιο εξειδικευμένη σύνδεση χρησιμοποιούμε τα pins 0 και 1 της πλακέτας. Γι' αυτό το λόγο αποφεύγουμε να κάνουμε χρήση των συγκεκριμένων pin κατά την διάρκεια διαφόρων εφαρμογών. Συντάσσοντας την εντολή **Serial.begin(BaudRate)** μέσα στην διαδικασία **Void Setup()** δίνουμε την ευκαιρία στον μικροελεγκτή μας να ενεργοποιήσει την σειριακή θύρα επικοινωνίας του. Το **BaudRate** μέσα στην παρένθεση αντιπροσωπεύει τον ρυθμό που γίνεται η μετάδοση των bits. Η πιο συχνή τιμή που χρησιμοποιούμε σ' αυτή την εντολή είναι η τιμή 9600.

Παράδειγμα **Serial.begin**:

```
Serial.begin(9600);
```

Αν θελήσουμε να επιτύχουμε αμφίδρομη επικοινωνία στις εφαρμογές έχουμε την δυνατότητα να κάνουμε επίσης χρήση της σειριακής θύρας, στέλνοντας και λαμβάνοντας δεδομένα. Συχνό φαινόμενο στον προγραμματισμό θεωρείται το **debugging**. Είναι ένας τρόπος επικοινωνίας ο οποίος μας δίνει την ευκαιρία να ελέγχουμε και να βλέπουμε τι τιμές παίρνουν οι μετρητές μας και οι μεταβλητές σε μία σειριακή οθόνη. Η εντολή που το πραγματοποιεί είναι η **print()** όπου κάνει εκτύπωση ένα μήνυμα ή διάφορες τιμές. Και η **println()** που την χρησιμοποιούμε για τον ίδιο λόγο ακριβώς με την μόνη διαφορά πως κάθε φορά που εκτυπώνει το αποτέλεσμα αλλάζει γραμμή.

Παράδειγμα **Serial.print()**:

```
Serial.print(" Η επικοινωνία έχει ξεκινήσει"); /* Αυτή η εντολή θα εμφανίσει το συγκεκριμένο μήνυμα στην οθόνη χωρίς να πραγματοποιήσει αλλαγή γραμμής. */
```

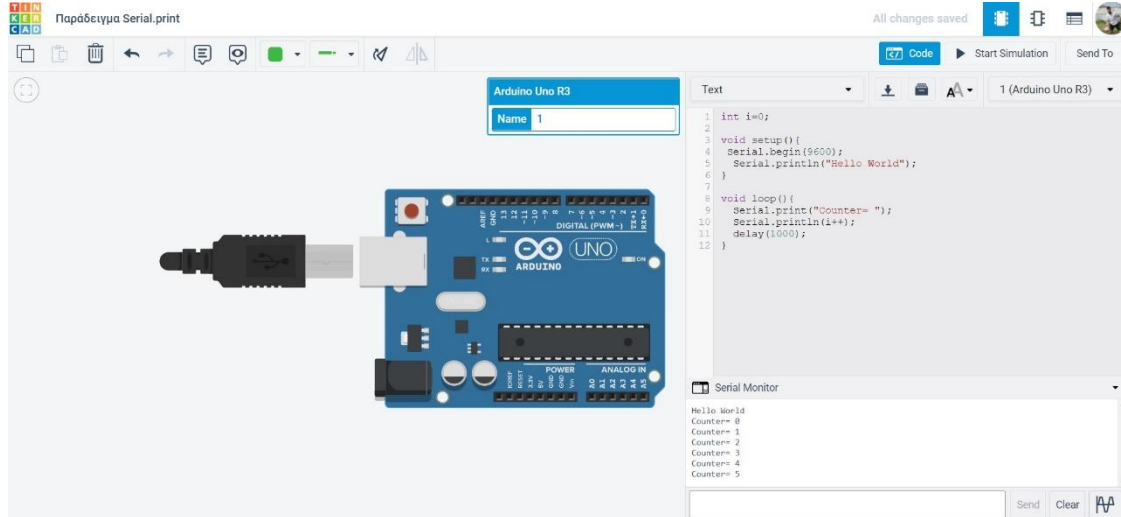
Παράδειγμα **Serial.println()**:

```
Serial.println(value); /* Αυτή η εντολή θα εμφανίσει το περιεχόμενο της
```

Διπλωματική Εργασία



μεταβλητής value σε μια γραμμή. */



Εικόνα 2- Παράδειγμα Serial.print στην πλατφόρμα Tinkercad

3.2.7 Η δομή επιλογής στο Arduino (If)


Στον προγραμματισμό, συχνό φαινόμενο αποτελεί η διαδικασία λήψης αποφάσεων. Όπου με την βοήθεια μίας συνθήκης επιλέγουμε και ελέγχουμε αν θα εκτελεστεί και σε πιο σημείο το πρόγραμμα μας. Η δομή επιλογής μας δίνει αυτή την δυνατότητα και συντάσσετε με τον παρακάτω τρόπο:

```
If <συνθήκη> { <Εντολές 1> }
Else { <Εντολές 2> }
/* Στην συνθήκη πραγματοποιείται ο έλεγχος που θέλουμε να γίνει και συνήθως
χρησιμοποιούμε τους τελεστές σύγκρισης. */
```

Οι τελεστές σύγκρισης που χρησιμοποιούμε στο Arduino είναι:

- >: Το μεγαλύτερο
- <: Το μικρότερο
- =: Το ίσο
- <>: Το διάφορο
- >=: Το μεγαλύτερο ίσο



 <=: Το μικρότερο ίσο

Επίσης η συνθήκη που θα χρησιμοποιήσουμε στο πρόγραμμα μας υπάρχει η πιθανότητα να είναι σύνθετη. Γι' αυτό το λόγο μπορούμε να χρησιμοποιήσουμε τους λογικούς τελεστές στο Arduino που τους συντάσσουμε με τον εξής τρόπο:

 &&: Λογικό ΚΑΙ

 ||: Λογικό Η

 NOT: Λογικό ΟΧΙ

Αφού πραγματοποιηθεί ο έλεγχος της συνθήκης και είναι αληθής θα εκτελεστούν οι εντολές 1 που βρίσκονται μέσα στις αγκύλες του παραδείγματος που δώσαμε. Αν η συνθήκη είναι ψευδής θα πραγματοποιηθούν οι εντολές 2 που βρίσκονται μετά το else. Ωστόσο δεν είναι απαραίτητο κατά την δομή επιλογής να υπάρχει το else. Υπάρχουν και πολυπλοκότερες εντολές για την δομή επιλογή στο Arduino, αλλά δεν θ' αναφερθούμε στον οδηγό αυτό διότι απευθύνεται σε αρχαρίους.




3.2.8 Η δομή επανάληψης στο Arduino (For-While)

37

Στον προγραμματισμό κάποιες φορές είναι επιτακτική ανάγκη να πραγματοποιήσουμε μία διαδικασία για αρκετές φορές. Έτσι χρησιμοποιώντας την δομή επανάληψης έχουμε την δυνατότητα να κάνουμε επανάληψη σ' ένα σύνολο από εντολές καταμετρώντας τις επαναλήψεις, ή μπορούμε να ελέγχουμε κάθε φορά μία συνθήκη για να υλοποιήσουμε τις επαναλήψεις που ζητάμε. Η πιο συχνή μορφή επανάληψης που συναντάμε στο Arduino είναι αυτή που γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων. Ο αριθμός είναι προκαθορισμένος και η εντολή συντάσσεται με τον εξής τρόπο:

```
For (<Αρχική Τιμή>;<Συνθήκη_Τερματισμού>;Βήμα>
{ <Εντολές> }
```

Οπότε κάνουμε χρήση μίας μεταβλητής ελέγχου:

-  Αρχική τιμή: Δίνουμε στην μεταβλητή μας μια αρχική τιμή. (π.χ $i=0$)
-  Βήμα: Εδώ κρατάμε τις αλλαγές που πραγματοποιούνται σε κάθε επανάληψη (π.χ $i+3$ ή $i++$ (το οποίο σημαίνει κάθε φορά να αλλάζει βήμα κατά 1 $i+1$)).
-  Συνθήκη τερματισμού: Εδώ τοποθετούμε την συνθήκη στην οποία γίνεται ο έλεγχος για να τερματιστεί η επανάληψη.

Παράδειγμα δομής επανάληψης for:



```
for (i=1;i<15;i=i++) { // Η συνθήκη που αναφερθήκαμε πιο πάνω.  
  
brightness = brightness +5; //Στην συγκεκριμένη μεταβλητή προσθέτω 5.  
  
analogWrite(ledPin, brightness); //Κάνω μια αναλογική εκτύπωση στις 2 μεταβλητές.  
  
};
```

Στο Arduino επίσης έχουμε την δυνατότητα να χρησιμοποιήσουμε εντολές επανάληψης στις οποίες δεν είναι προκαθορισμένος ο αριθμός επαναλήψεων. Παρόλα αυτά και αυτές έχουν μια συνθήκη που γίνεται έλεγχος και συντάσσονται με τον εξής τρόπο:

```
While <συνθήκη> { <εντολές> } /*Στην συγκεκριμένη εντολή όσο ισχύει η συνθήκη  
εκτελούνται οι εντολές. */
```

Παράδειγμα επανάληψης While:

```
while (cnt < 5) { /* Η επανάληψη εκτελείται όσο η μεταβλητή είναι μικρότερη του  
5. */  
  
digitalWrite(ledpin, HIGH); /* Πραγματοποιεί ψηφιακή εκτύπωση στην μεταβλητή  
ledpin και της δίνει την μέγιστη τιμή. */  
  
delay(200); // Πραγματοποιεί καθυστέρηση για 200 milliseconds.  
  
digitalWrite(ledpin, LOW); /* Πραγματοποιεί ψηφιακή εκτύπωση στην μεταβλητή  
ledpin και της δίνει την ελάχιστη τιμή. */  
  
delay(200); // Πραγματοποιεί καθυστέρηση για 200 milliseconds.  
  
cnt++;} //Αυξάνει την μεταβλητή κατά 1.
```

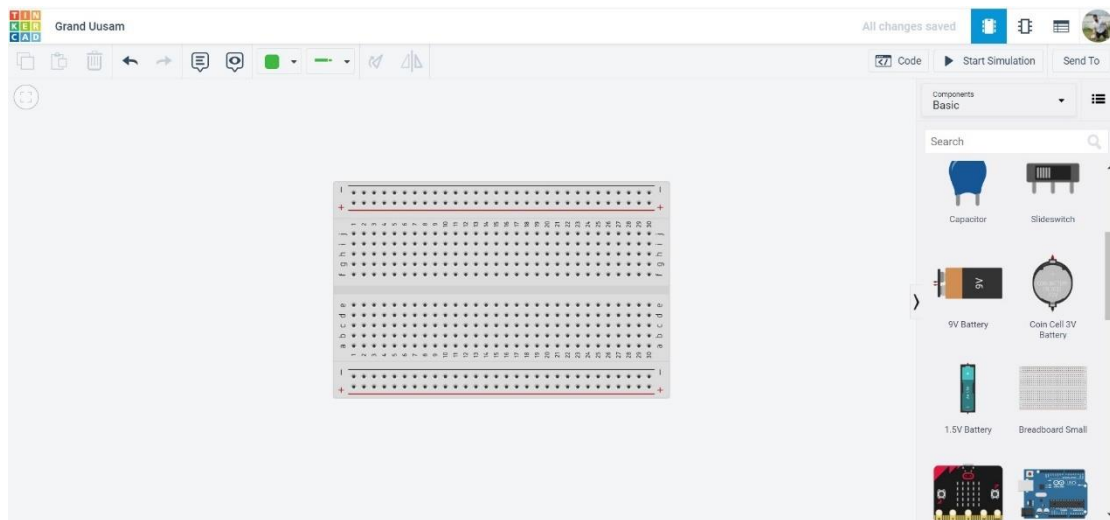
3.2.9 Η χρήση του Breadboard στο Arduino

Κατά την δημιουργία διάφορων κυκλωμάτων και εφαρμογών στο Arduino είναι επιτακτική ανάγκη να χρησιμοποιήσουμε διάφορα εξαρτήματα. Αυτά τα εξαρτήματα συνήθως είναι leds, αντιστάσεις, αισθητήρες, καλώδια κ.α. Υπάρχει η δυνατότητα η

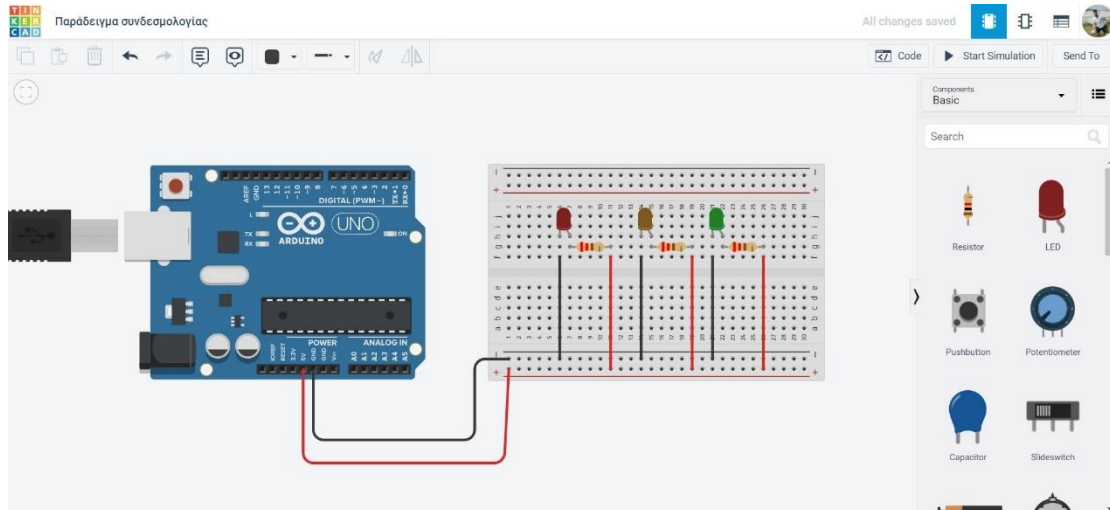


διασύνδεση τους να υλοποιηθεί με τα καλώδια μόνο, ωστόσο αν χρησιμοποιήσουμε πολλά εξαρτήματα τα κυκλώματα γίνονται αρκετά πολύπλοκα. Για τον λόγο αυτό συνηθίζουμε να χρησιμοποιούμε το breadboard. Το Breadboard μας παρέχει την δυνατότητα να κάνουμε βραχυκυκλώσεις μεταξύ των εξαρτημάτων, χωρίς την χρήση των καλωδίων. Κατά συνέπεια χρησιμοποιούμε λιγότερα καλώδια και τα κυκλώματα γίνονται πιο απλά.

Θα παρατηρήσουμε πως στις πλευρές πάνω και κάτω υπάρχουν τα σύμβολα του + και του -. Εκεί τοποθετείται συνήθως η τροφοδοσία και η γείωση από τους ακροδέκτες του μικροελεγκτή μας. Στο - τοποθετείται το GND(γείωση) και στο + τοποθετείται η τροφοδοσία. Το + και το - είναι οριζόντια βραχυκυκλωμένο έτσι αν τοποθετηθεί τροφοδοσία ή γείωση παίρνουν όλα τα pins. Ενώ τα pins τα οποία βρίσκονται εσωτερικά είναι βραχυκυκλωμένα κάθετα μεταξύ τους ανά πεντάδες. Οποιαδήποτε γέφυρα χρειαστούμε μπορούμε να την κάνουμε με κάποιο καλώδιο η ακόμη και με κάποιο εξάρτημα.



Εικόνα 3- Παράδειγμα Breadboard



Εικόνα 4-Παράδειγμα συνδεσμολογίας με την βοήθεια του Breadboard

Θα παρατηρήσουμε σύμφωνα με την παραπάνω εικόνα πως όλα τα εσωτερικά pins είναι βραχυκυκλωμένα κάθετα και αν θέλω να πραγματοποιήσω κάποια άλλη συνδεσμολογία, μπορώ να χρησιμοποιήσω καλώδια ή εξαρτήματα.

3.2.10 Τα κυριότερα εξαρτήματα που θα χρησιμοποιήσει ένας αρχάριος

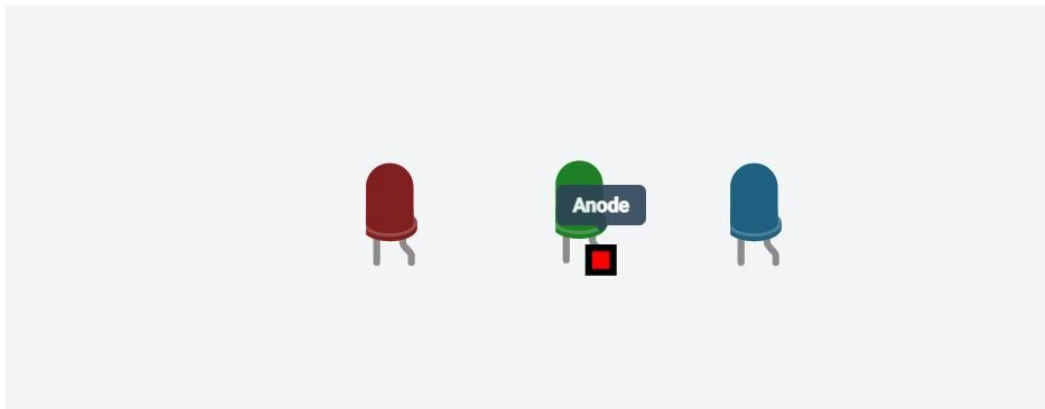
Το Arduino γενικότερα περιέχει μεγάλη γκάμα σε εξαρτήματα τα οποία μπορούμε να κάνουμε χρήση σε διάφορες εφαρμογές. Εμείς εδώ θα αναλύσουμε τα κυριότερα εξαρτήματα τα οποία θα χρειαστεί ένας αρχάριος για να ολοκληρώσει τα φύλλα εργασίας που θα δούμε παρακάτω. Η δυσκολία των φύλλων εργασίας θα πηγαίνει κλιμακωτά και θα χρησιμοποιείται το ανάλογο εξάρτημα αφού έχει αναλυθεί.

Οι φωτοдиодοι-Leds

Τα τελευταία χρόνια τα Leds έχουν επικρατήσει στην αγορά και χρησιμοποιούνται όχι μόνο σε εφαρμογές, αλλά και σε πολλές συσκευές. Παρέχονται σε διάφορα χρώματα και για να τα κάνουμε χρήση τους θα πρέπει να γίνει η σωστή επιλογή στην αντίσταση, ειδάλλως καταστρέφονται μετά από λίγο. Αρκεί να κάνουμε χρήση του γνωστού μας τύπου $R = (V_{\pi} - V_{\lambda})/I$ για να πραγματοποιηθεί η σωστή επιλογή στην αντίσταση. Αναλύοντας τον τύπο θα δούμε ότι το R είναι η αντίσταση, το V_{π} η τάση που παρέχεται από την πηγή, το V_{λ} η τάση λειτουργίας του Led και το I το ρεύμα



λειτουργίας του Led. Η αντίσταση που συνηθίζουμε να χρησιμοποιούμε στα περισσότερα Leds είναι της τάξεως των 220 Ω πλην εξαιρετικών περιπτώσεων. Όπως θα παρατηρήσετε στην παρακάτω φωτογραφία θα διακρίνεται πως τα Led διαθέτουν πολικότητα. Αυτό σημαίνει πως στο μέρος της ανόδου κάνουμε σύνδεση το θετικό πόλο της τροφοδοσίας μας (+), ενώ στην κάθοδο συνδέουμε την γείωση (-). Σε περίπτωση που η σύνδεση πραγματοποιηθεί με διαφορετικό τρόπο απλά δεν θ ανάψει το Led.



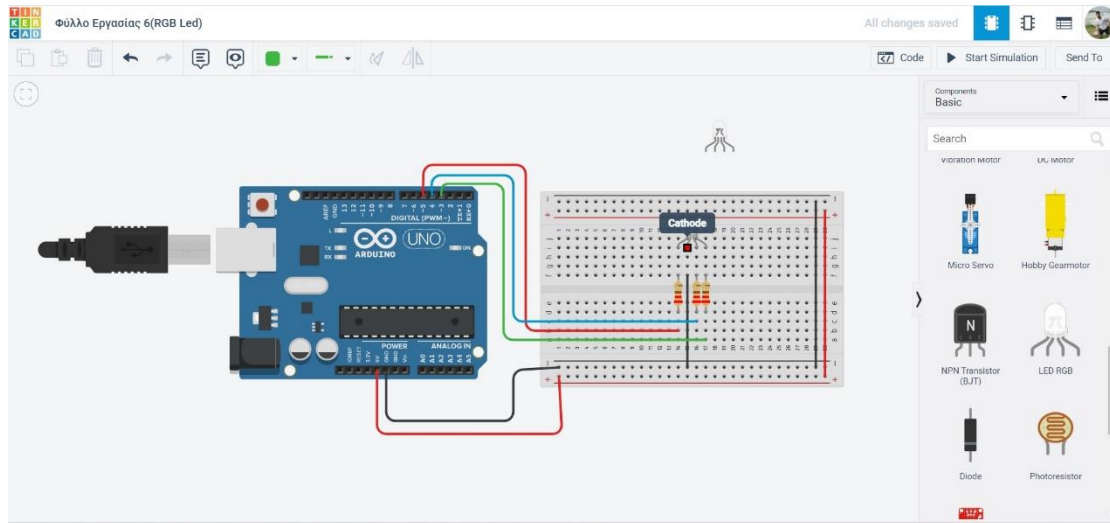
Εικόνα 5- Παράδειγμα από Leds στο Tinkercad

RGB LEDs

Τα RGB LEDs μοιάζουν αρκετά με τα LEDs που περιγράψαμε πιο πάνω με την διαφορά πως βασίζονται σε συγκεκριμένα χρώματα και η συνδεσμολογία τους διαφέρει λιγάκι από ένα απλό led. Καταρχάς το όνομα RGB προέρχεται από τα χρώματα που μπορεί να πάρει το Led (Red, Green, Blue). Όπως θα δούμε και στην φωτογραφία παρακάτω μπορούμε να το παρομοιάσουμε ως 3 Led σε 1. Τα πόδια διασύνδεσης είναι 4. Αντιστοιχεί ένα για κάθε χρώμα καθώς και ένα ποδαράκι είναι για την γείωση. Παράλληλα όπως κάναμε και στο απλό Led έτσι και στο RGB θα πρέπει σε κάθε ποδαράκι που αντιστοιχεί σε χρώμα να προστεθεί αντίσταση. Η τιμή της αντίστασης βγαίνει από τον τύπο που αναφέραμε πριν. Ωστόσο αν χρησιμοποιήσουμε μια των 180 Ω για το κόκκινο χρώμα και κοντά στα 100 Ω για τα άλλα δύο θα είμαστε λειτουργικοί. Γενικότερα αν δεν είμαστε σίγουροι για το τι αντίσταση θα πρέπει να χρησιμοποιήσουμε για όλους τους αισθητήρες, επιλέγουμε πάντα μεγαλύτερη τιμή. Έτσι το χειρότερο που μπορούμε να κάνουμε είναι να μην λειτουργήσει το αισθητήριο που χρησιμοποιούμε, ενώ αν βάλουμε μικρότερη από ότι πρέπει θα καεί από υπέρταση. Την συνδεσμολογία θα την δούμε στην παρακάτω



εικόνα. Το πρώτο πόδι από τα αριστερά αντιστοιχεί στο κόκκινο χρώμα, το δεύτερο από τα αριστερά στην γείωση, το τρίτο στο πράσινο και το τέταρτο στο μπλε.

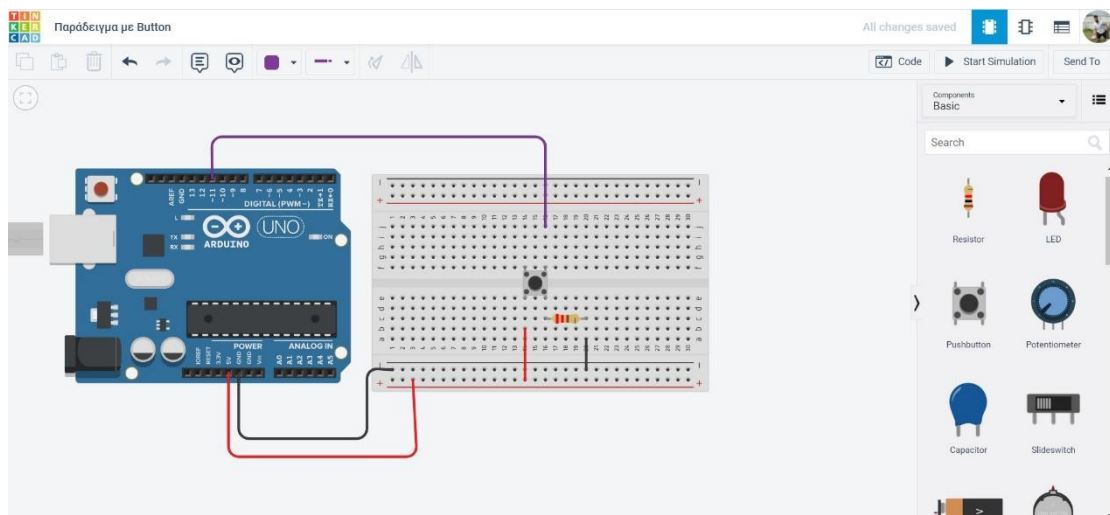


Εικόνα 6- Παράδειγμα RGB στην πλατφόρμα Tinkercad

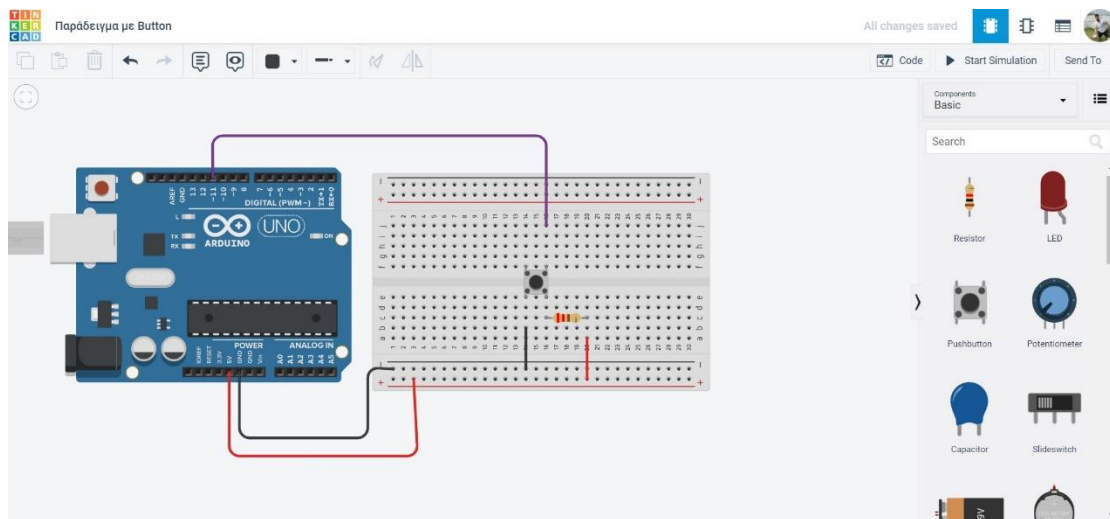
Τα κουμπιά (Buttons)

Τα Button είναι ακόμη ένα εξάρτημα το οποίο θα κληθούμε να χρησιμοποιήσουμε στην πλατφόρμα του Arduino. Μας δίνουν την ευκαιρία να παρέμβουμε στο κύκλωμα όπως να δώσουμε ρεύμα σε κάποιο τμήμα του κυκλώματος, ή να κάνουμε διακοπή της τροφοδοσίας. Παρακάτω θα σας παραθέσω δύο παραδείγματα με κυκλώματα από button και θα εξηγήσουμε την λειτουργία τους.

42



Εικόνα 7- Παράδειγμα Pull down resistor στο Tinkercad



Εικόνα 8- Παράδειγμα Pull up resistor στο Tinkercad

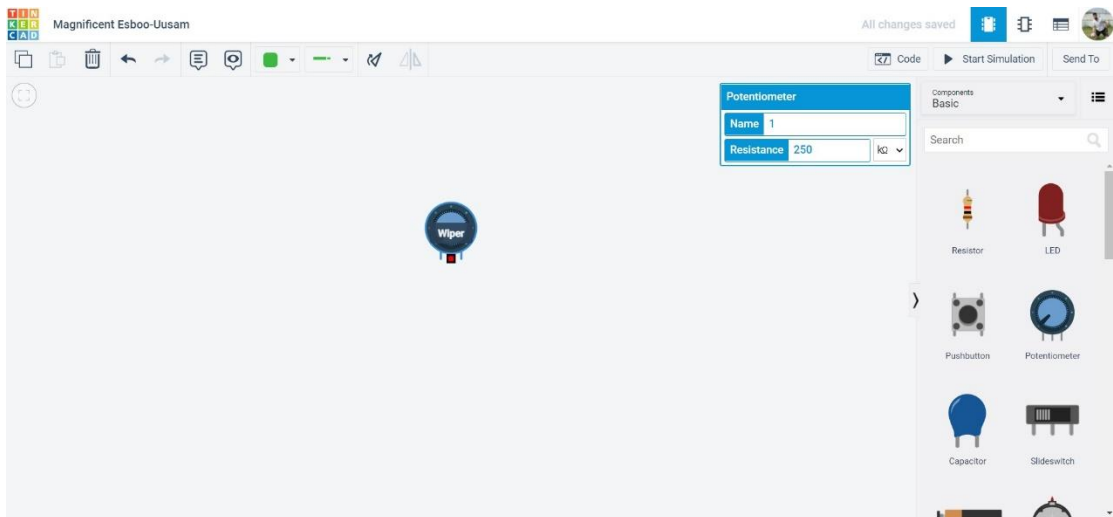
- ✚ Pull down resistor: Το pull down resistor λειτουργεί με το εξής τρόπο. Η θύρα εισόδου μένει μόνιμα στην τιμή LOW και μόλις πατηθεί το button καταλήγει στο HIGH. Το κύκλωμα έχει απλή συνδεσμολογία καθώς η γείωση πηγαίνει στο κάτω αριστερό ποδαράκι από το button. Η τάση μαζί με την αντίσταση στο κάτω δεξί ποδαράκι και στο πάνω δεξί ποδαράκι συνδέεται στην θύρα του Arduino.
- ✚ Pull up resistor: Το Pull up resistor έχει ακριβώς το ίδιο κύκλωμα με το pull down resistor, με την μόνη διαφορά πως γίνεται η αντιστροφή της τροφοδοσίας με την γείωση. Αυτό έχει ως αποτέλεσμα το button που χρησιμοποιούμε να λειτουργεί ακριβώς με τον αντίστροφο τρόπο. Δηλαδή το input pin έχει πάντα την τιμή HIGH και αλλάζει μόνιμα σε LOW μόλις πατηθεί το button.

Τα button που χρησιμοποιούμε στο Arduino έχουν μέσα τους μία μικρή αντίσταση. Παραδοσιακά το ρεύμα ακολουθεί την διαδρομή που έχει την μικρότερη αντίσταση στο κύκλωμα μας. Κατά συνέπεια στο Pull up resistor όταν πιεστεί το button κλείνει και το κύκλωμα μας, και από τα 5V καταλήγει στην γείωση. Στο Pull down resistor όταν πιεστεί το button πραγματοποιείται η ροή του ρεύματος από την πηγή προς τη είσοδο και αυτό έχει ως αποτέλεσμα να ενεργοποιείται το κύκλωμα.



Το ποτενσιόμετρο

Το ποτενσιόμετρο είναι ένα αναλογικό εξάρτημα το οποίο χρησιμοποιούμε πολύ συχνά στις εφαρμογές που υλοποιούμε στο Arduino. Στην ουσία λογίζεται ως μια αντίσταση η οποία μεταβάλλεται. Οπότε γυρνώντας το κουμπί πραγματοποιούμε αυξομείωση της αντίστασης, και ανάλογα που θα το μετακινήσουμε το ρεύμα αυξάνεται ή μειώνεται. Περιέχει τρία ποδαράκια διασύνδεσης όπου στο ένα συνδέουμε την γείωση και στο άλλο την τροφοδοσία. Στο μεσαίο ποδαράκι συνδέεται η εντολή που μεταβάλλει την τιμή του ρεύματος (Αναλογική σύνδεση στο Arduino.). Έτσι θεωρείται ένας διαιρέτης τάσης όπου μεταβάλλεται κάθε φορά που περιστρέφουμε το κουμπί του. Στα παρακάτω φύλλα εργασία θα δούμε πως έχουμε την δυνατότητα να το χρησιμοποιήσουμε στα Led για να ρυθμίσουμε την ένταση του φωτός, καθώς και στην αυξομείωση του ήχου κ.α .



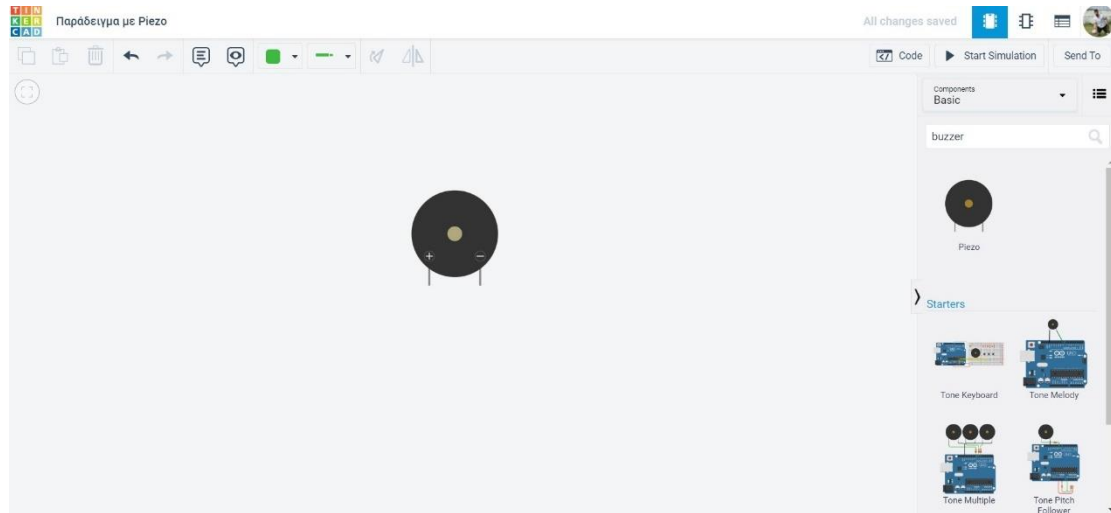
Εικόνα 9- Παράδειγμα με ποτενσιόμετρο στο Tinkercad

Το Ηχείο (Sounder-Piezo)

Ακόμη ένα εξάρτημα το οποίο χρησιμοποιούμε πολύ συχνά στις εφαρμογές του Arduino αλλά και γενικότερα στην καθημερινή μας ζωή είναι το ηχείο. Η λειτουργία του μοιάζει αρκετά με εκείνη του Led, δηλαδή η συνδεσμολογία του γίνεται με δύο καλώδια. Το ένα είναι η γείωση και το άλλο είναι η τροφοδοσία των 5V. Αν η τροφοδοσία που θα δώσουμε είναι σταθερή (5V) ο ήχος που θα μας δώσει είναι σταθερός, ενώ αν η τάση μας είναι μεταβλητή και η συνδεσμολογία πραγματοποιηθεί σε ένα αναλογικό Pin (PWM) ο ήχος μας θα είναι μεταβλητός. Το εξάρτημα που θα



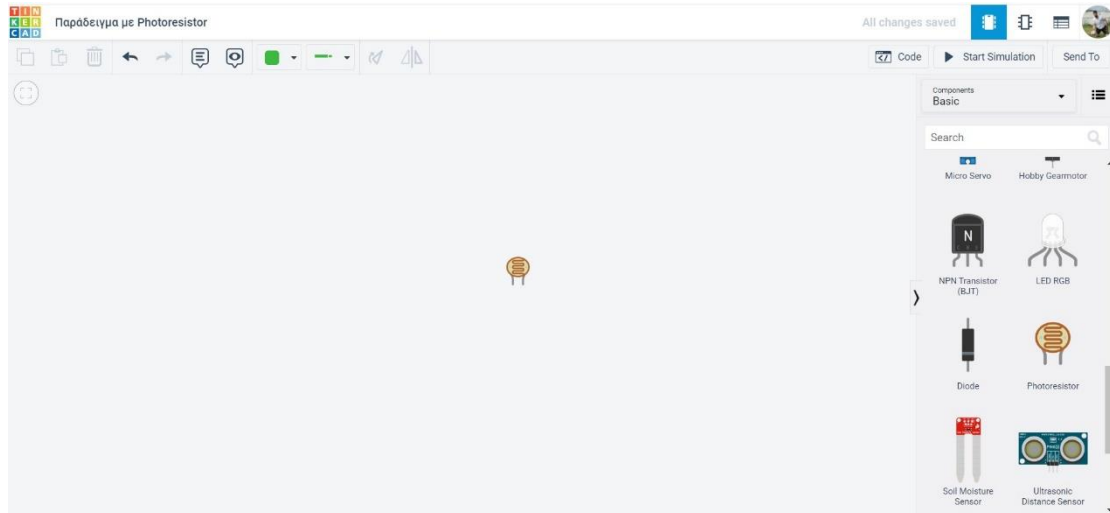
χρησιμοποιήσουμε εμείς είναι το ηχείο (Piezo). Το χαρακτηριστικό του είναι πως ανάλογα με το πόσο το πιέζουμε τροποποιούνται οι ιδιότητες του. Παρακάτω θα παραθέσουμε παράδειγμα με το συγκεκριμένο εξάρτημα.



Εικόνα 10- Παράδειγμα με ηχείο (Piezo) στο Tinkercad

Η φωτοαντίσταση (Photoresistor)

Η αντίσταση που μεταβάλλεται αναλογικά με το φως το οποίο δέχεται ονομάζεται φωτοαντίσταση. Όταν το φως που διαρρέει την αντίσταση είναι μεγάλης φωτεινότητας η αντίσταση της φωτοαντίστασης μικραίνει, ενώ όταν το περιβάλλον είναι πιο σκοτεινό η αντίσταση μεγαλώνει. Η σύνδεση της πραγματοποιείται σε κύκλωμα το οποίο έχει σταθερή τάση, και το ρεύμα θα μεταβληθεί εφόσον μεταβληθεί και η φωτεινότητα που την διαρρέει. Τα κυκλώματα που είναι βασισμένα σε μία φωτοαντίσταση παίρνουν ως είσοδο την φωτεινότητα και λειτουργούν ανάλογα με την μεταβολή της. Χρησιμοποιούνται αρκετά στις εφαρμογές στο Arduino αλλά και στην καθημερινή μας ζωή, όπως στην ανίχνευση της κίνησης σε κάποιο χώρο.



Εικόνα 11- Παράδειγμα φωτοαντίστασης στο Tinkercad

Ο αισθητήρας υπερήχων (Ultrasonic sensor)

Ο τελευταίος αισθητήρας που θα ασχοληθούμε και έπειτα θα περάσουμε στα φύλλα εργασίας είναι ο αισθητήρας υπερήχων. Η λειτουργία του αισθητήρα είναι λίγο διαφορετική σε σχέση με τους άλλους αισθητήρες που έχουμε δει μέχρι τώρα. Έτσι κάνει χρήση δύο διατάξεων, την μία για να στείλει ένα υπερηχητικό κύμα και την άλλη για να το λάβει. Οπότε ο αισθητήρας εκμεταλλεύεται το χρόνο που μεσολαβεί σ' αυτή την αποστολή και παραλαβή των σημάτων και μας δίνει την δυνατότητα να κάνουμε τον υπολογισμό της απόστασης, από ένα αντικείμενο που έχουμε μπροστά μας. Η συνδεσμολογία του είναι σχετικά απλή καθώς περιέχει 4 ποδαράκια διασύνδεσης. Το Vcc είναι για την τροφοδοσία και τα Volt που χρησιμοποιεί είναι 5. Στο ποδαράκι GND συνδέουμε την γείωση. Στο Trig(Trigger) πραγματοποιούμε σύνδεση με ένα ψηφιακό pin του Arduino για την αποστολή του σήματος. Ενώ στο ποδαράκι Echo γίνεται διασύνδεση με ένα ψηφιακό pin για την παραλαβή του σήματος που επιστρέφει. Η δήλωση των pin γίνεται μέσα στην Void Setup() και το καλώδιο που έχει διασύνδεση στο Trigger δηλώνεται ως έξοδο, ενώ το echo ως είσοδο. Όταν θέλουμε να κάνουμε αποστολή ενός σήματος απλά μπορούμε να δώσουμε τάση στο ποδαράκι που βρίσκεται το trigger. Θα ακολουθήσει παράδειγμα το οποίο θα περιέχει όλες τις εντολές που αφορούν τον υπολογισμό της απόστασης αλλά και την επιστροφή του σήματος.



Σεϊτανίδης Γεώργιος, Σχεδιασμός και σύνταξη αναλυτικών οδηγιών χρήσης για Arduino και Micro:bit

Κατασκευαστικά από τον αισθητήρα μας δίνεται ότι η τιμή που δίνεται για τον υπολογισμό της απόστασης είναι 58.138 και προκύπτει από την ταχύτητα του ήχου. Μία πιο λεπτομερής παρατήρηση θα μας δώσει σίγουρα καλύτερες τιμές διότι οι μετρήσεις μας συνήθως είναι εμπειρικές.

Ο τύπος που μας δίνει την απόσταση είναι ο εξής:

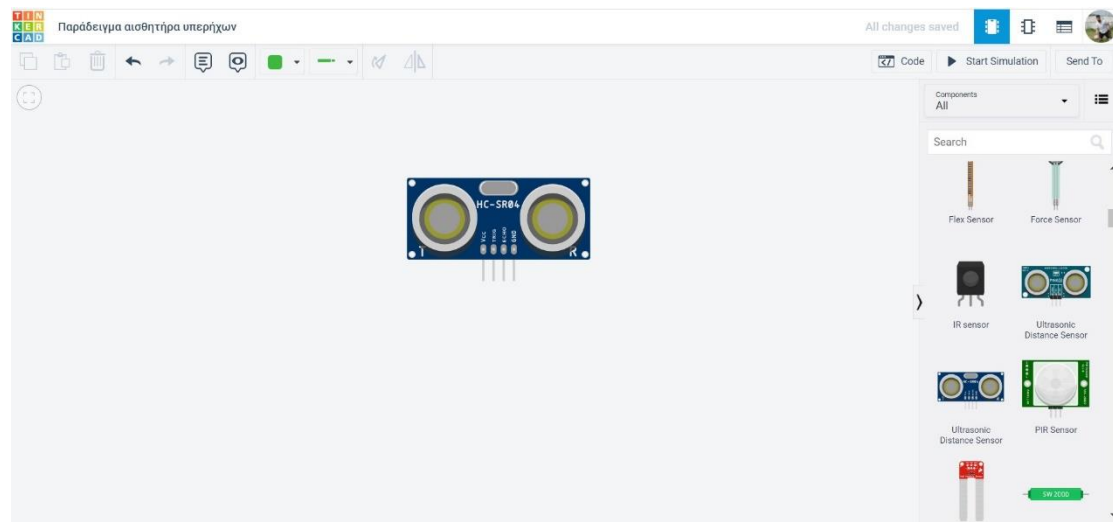
$$\text{Distance} = ((\text{Η διάρκεια του παλμού HIGH}) * (\text{Η ταχύτητα του ήχου (340m/s.)}) / 2$$

✚ Η διαίρεση με το 2 προκύπτει διότι το σήμα πραγματοποιεί διπλάσια απόσταση για να πάει και να έρθει.

Παράδειγμα αισθητήρα υπερήχων:

```
digitalWrite(trigPin, HIGH); /* Δίνουμε τάση στο pin του trigger για να ξεκινήσει η αποστολή του σήματος. */  
delayMicroseconds(10) // Δίνω λίγο χρόνο για να πραγματοποιηθεί η αποστολή  
digitalWrite(trigPin, LOW); /* Σταματάω την αποστολή μηνύματος μηδενίζοντας την τάση. */  
duration = pulseIn(echoPin, HIGH); /* Κρατάμε τον χρόνο που έχει περάσει έως να επιστραφεί το σήμα στο echo. */  
distance = duration/58.138; /* Πραγματοποιείται ο υπολογισμός της απόστασης σύμφωνα με την ταχύτητα του ήχου. */
```

47



Εικόνα 12- Παράδειγμα αισθητήρα υπερήχων στο Tinkercad

Διπλωματική Εργασία



3.3 Φύλλα εργασίας για αρχάριους στο Arduino

Παρακάτω θα παρατεθούν κάποια φύλλα εργασίας για μαθητές όλων των τάξεων. Θα περιέχουν αναλυτικές οδηγίες για τον αισθητήρα που θα χρησιμοποιούν κάθε φορά. Η δυσκολία θα οδηγείται κλιμακωτά από το ένα φύλλο εργασίας στο άλλο. Θα παρέχεται επίσης το κύκλωμα για το αισθητήρα που αναφέρουμε καθώς και ο κώδικας υλοποίησης. Θα ακολουθήσουμε την βασική αρχή στον προγραμματισμό όπου υλοποιώντας απλά φύλλα εργασίας, θα οδηγηθούμε στα πιο σύνθετα. (Τεμαχισμός σύνθετου προβλήματος σε πολλά απλούστερα προβλήματα.) Τέλος αφού ολοκληρωθούν όλα τα φύλλα εργασίας θα ακολουθήσουν εκπαιδευτικά σενάρια μεγαλύτερης δυσκολίας.

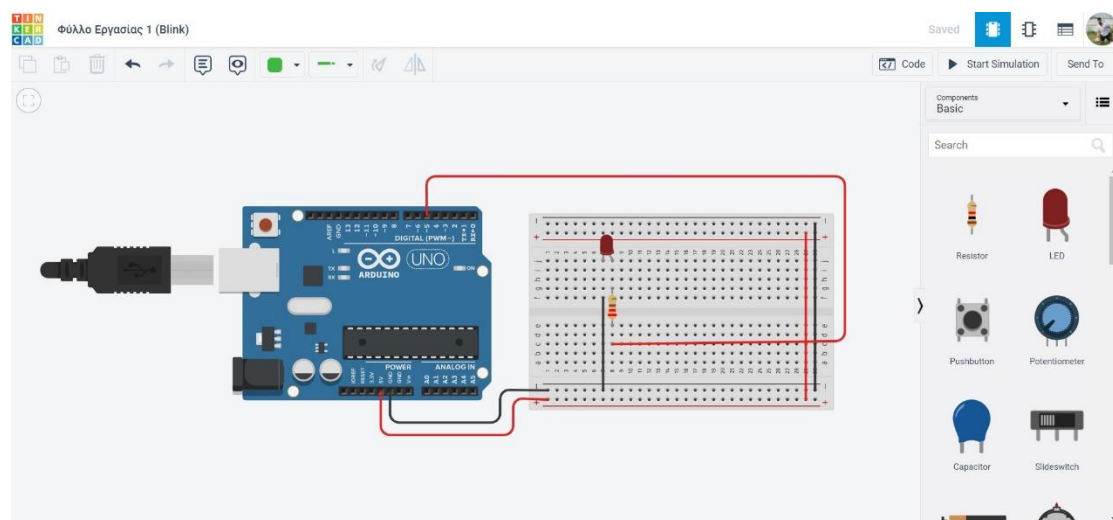
3.3.1 Φύλλο εργασίας 1- Το Led που αναβοσβήνει (Blink)

Το πρώτο φύλλο εργασίας αφορά ένα Led όπου με το κατάλληλο κύκλωμα και κώδικα θα το κάνουμε να αναβοσβήνει. Αυτό θα συμβαίνει ανά ένα δευτερόλεπτο και θα επαναλαμβάνεται συνέχεια μέχρις ότου να μην δέχεται ρεύμα το Arduino.

48

Περιγραφή του κυκλώματος

Για την εφαρμογή μας θα χρειαστούμε ένα Led ό,τι χρώματος εμείς θέλουμε και μια αντίσταση των 220Ω. Η αντίσταση τοποθετείτε για την ασφάλεια του Led καθώς άμα κάνει υπέρταση (5V και πάνω) το Led θα καεί. Παρακάτω θα παραθέσουμε την συνδεσμολογία του κυκλώματος.



Εικόνα 13- Παράδειγμα Led Blink στο Tinkercad



Το Pin5 θεωρείται για εμάς έξοδος και η σύνδεση της αντίστασης είναι μεταξύ της συγκεκριμένης εξόδου και του Led. Ο σκοπός της τοποθέτησης της αντίστασης είναι για την προστασία του Led για να μην καεί. Στο στραβό ποδαράκι του Led όπως βλέπουμε στο σχήμα συνήθως κάνουμε σύνδεση την τροφοδοσία, ενώ στο άλλο πόδι συνδέουμε την τροφοδοσία.

Ο προγραμματισμός του κυκλώματος

Για ευκολία θα δώσουμε τις οδηγίες για το πως πρέπει να γίνει η σύνδεση του κυκλώματος:

1. Δίνουμε μία μεταβλητή που μ' αυτή θα παρομοιάζουμε το Pin μας, οπότε στην συγκεκριμένη περίπτωση δίνουμε την τιμή 5.
2. Δηλώνουμε το Pin5 ως έξοδο.
3. Δίνουμε τροφοδοσία (5V) στο Pin5.
4. Αναμένουμε 1 δευτερόλεπτο μέσα στο οποίο θα ανάψει το Led μας.
5. Πραγματοποιούμε διακοπή της τροφοδοσίας (0V) στο Pin5.
6. Αναμένουμε 1 δευτερόλεπτο μέσα στο οποίο θα σβήσει το Led μας.

Η διαδικασία των βημάτων 1,2 θα πραγματοποιηθούν μόνο μία φορά, ενώ τα βήματα 3 και 6 θα πρέπει να γίνονται επανάληψη συνέχεια.

Οι έννοιές που μαθαίνω: Θύρες, Μεταβλητές, Διαχείριση των θυρών, Ψηφιακή έξοδος, Καθυστέρηση, Breadboard, Led, Φόρτωση.

Ο κώδικας υλοποίησης

```
int ledPin = 5;  
  
void setup() {  
  
pinMode(ledPin, OUTPUT);
```



```
}  
void loop()  
{  
  
digitalWrite(ledPin, HIGH);  
delay(1000);  
digitalWrite(ledPin, LOW);  
delay(1000);  
}
```

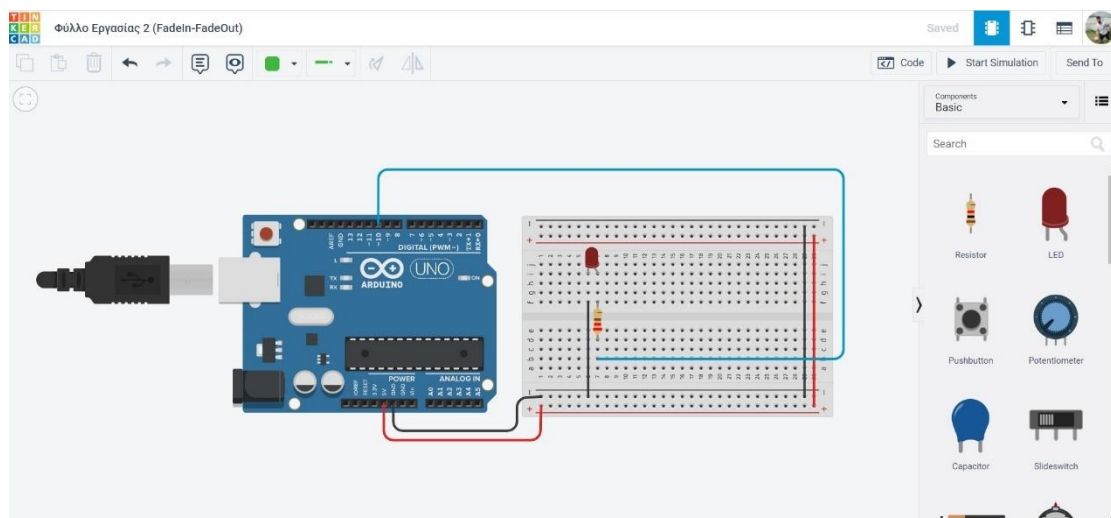
3.3.2 Φύλλο εργασίας 2- Η σταδιακή αύξηση και μείωση της φωτεινότητας στο Led (Fade in – Fade out)

Στο συγκεκριμένο φύλλο εργασίας θα χρησιμοποιήσουμε ξανά ένα Led με την διαφορά πως το Led δεν θα παίρνει τις μέγιστες τιμές, αλλά στην ένταση του θα υπάρχει μία αυξομείωση. Κατά συνέπεια το Led θα ξεκινά να ανάβει από πολύ μικρή ένταση και θα φτάνει στο μέγιστο της φωτεινότητάς του. Η διαδικασία αυτή έπειτα θα λειτουργεί αντίστροφα μέχρι να φτάσει η ένταση του στο χαμηλότερο επίπεδο της.

50

Περιγραφή του κυκλώματος

Για το συγκεκριμένο φύλλο εργασίας θα χρειαστούμε τα ίδια εξαρτήματα με το παραπάνω φύλλο, δηλαδή ένα Led και μία αντίσταση. Θα ακολουθήσει η συνδεσμολογία του κυκλώματος.



Εικόνα 14- Παράδειγμα FadeIn - FadeOut στο Tinkercad



Ο προγραμματισμός του κυκλώματος

Θα παρατηρήσουμε ότι η συνδεσμολογία των εξαρτημάτων με το προηγούμενο φύλλο εργασίας είναι κοινή. Η μόνη διαφορά είναι πως συνδέω την έξοδο μου σε μία από τις θύρες PWM (Ψευτοαναλογικές). Ο λόγος είναι πως οι τιμές που μπορούν να πάρουν είναι από το 0 έως και το 255, οπότε θα μπορέσουμε να προσομοιώσουμε τις τάσεις 0V έως 5V και τις ενδιάμεσες τιμές. Κατά συνέπεια θα προγραμματίσουμε την θύρα μας ως έξοδο και εν συνεχεία χρησιμοποιώντας την συνάρτηση analogWrite(Pin, Brightness), δίνοντας στην μεταβλητή Pin σταδιακά τις τιμές που επιθυμούμε από 0 έως 255. Να τονίσουμε πως δεν πρέπει να ξεπεραστούν οι συγκεκριμένες τιμές.

Για ευκολία θα δοθούν τα βήματα που πρέπει να ακολουθήσουμε:

1. Ορίζουμε μια μεταβλητή για την θύρα μας και την αρχικοποιούμε ως θύρα εξόδου.
2. Ορίζουμε μια μεταβλητή που αφορά την φωτεινότητα και της δίνουμε την αρχική τιμή 0.
3. Χρησιμοποιώντας την επανάληψη τμηματικά κάνουμε αύξηση της φωτεινότητας κατά 5 μέχρι να φτάσει η τιμή μου στο 255.
4. Όταν φτάσει η τιμή στο μέγιστο αναλογικό όριο (255) τότε την μηδενίζουμε και ξεκινάμε από την αρχή την διαδικασία.

51

Οι έννοιες που μαθαίνω: Αναλογική έξοδος.

Ο κώδικας υλοποίησης

```
int ledPin = 10;
int brightness = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (brightness == 255) {
    brightness = 0;
  };
  analogWrite(ledPin, brightness);
```



```
brightness = brightness + 5;  
delay(100);  
}
```

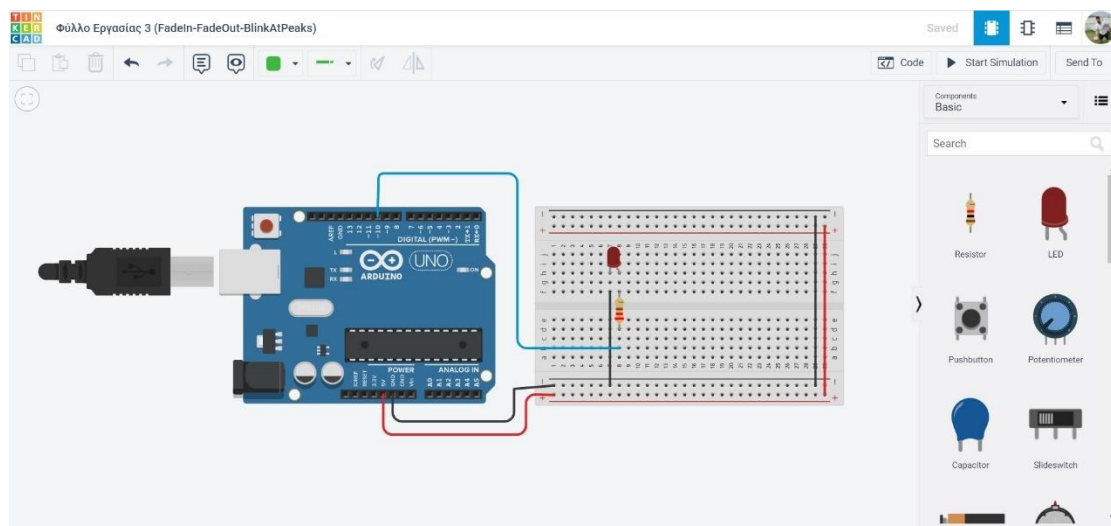
3.3.3 Φύλλο εργασίας 3 – Αυξομείωση της φωτεινότητας με αναβόσβημα στις άκρες (FadeIn – FadeOut με Blink στα Peaks)

Στο συγκεκριμένο φύλλο εργασίας το κύκλωμα παραμένει ακριβώς το ίδιο χωρίς καμία διαφορά. Η λειτουργία του περιέχει έναν συνδυασμό από τα δύο προηγούμενα. Η ένταση της φωτεινότητας ανεβαίνει σταδιακά και όταν φτάσει στο μέγιστο peak ανάβει και σβήνει το Led δύο φορές. Η αντίστοιχη διαδικασία συμβαίνει με τον ανάποδο τρόπο, όταν η φωτεινότητα του Led φτάνει στο ελάχιστο peak του με το Led μας να ανάβει και να σβήνει ξανά δύο φορές. Η διαδικασία επαναλαμβάνεται μέχρις ότου αποσυνδέσουμε την τροφοδοσία του Arduino.

Περιγραφή του κυκλώματος

Τα εξαρτήματα τα οποία θα χρησιμοποιήσουμε είναι ακριβώς τα ίδια με τα προηγούμενα φύλλα εργασίας, όπως και το κύκλωμα.

52



Εικόνα 15- Παράδειγμα FadeIn - FadeOut Blink at Peaks στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Για το συγκεκριμένο φύλλο εργασίας το κύκλωμα παραμένει το ίδιο και αυτό που αλλάζει είναι ο προγραμματισμός, για να υλοποιήσουμε το ζητούμενο:

Διπλωματική Εργασία



1. Πραγματοποιούμε τις δηλώσεις που είναι αναγκαίες αλλά και τις αρχικοποιήσεις που αφορούν τις τιμές, τις μεταβλητές και τις θύρες.
2. Ξεκινάμε την φωτεινότητα από την τιμή 0 και την ανεβάζουμε ανά 5 μέχρι να φτάσει 255 που είναι και το Peak.
3. Μόλις η τιμή μας θα φτάσει στο ανώτατο όριο (255) το Led μας θ ανάψει και θα σβήσει 2 φορές.
4. Ακολουθούμε την ανάποδη διαδικασία καθώς από την μέγιστη τιμή (255) μεταβάλλουμε προς τα κάτω ανά 5 μονάδες μέχρις ότου φτάσει στην χαμηλότερη τιμή της (0).
5. Φτάνοντας στην ελάχιστη φωτεινότητα το Led μας θ ανάψει και θα σβήσει 2 φορές και η διαδικασία θα επαναληφθεί μέχρις ότου αφαιρέσουμε την τροφοδοσία.

Οι έννοιες που μαθαίνω: Δομή επιλογής.

Ο κώδικας υλοποίησης

```
int ledPin = 10;
int brightness = 255;
int interval = 5; // διάστημα που θα ανεβαίνει ή κατεβαίνει η φωτεινότητα
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (brightness == 255 || brightness == 0) {
    // αναβοσβήνει (blink) δυο φορές
    digitalWrite(ledPin, HIGH);
    delay(100);
    digitalWrite(ledPin, LOW);
    delay(100);
    digitalWrite(ledPin, HIGH);
    delay(100);
    digitalWrite(ledPin, LOW);
    delay(100);
  }
}
```



```
/* αλλάζω πρόσημο στο διάστημα, ώστε αν ανέβαινε μέχρι τώρα(θετικό) να  
αρχίσει να κατεβαίνει(αρνητικό) κι αντίστροφα */  
interval = (-1)*interval;  
};  
analogWrite(ledPin, brightness);  
brightness = brightness + interval;  
delay(100);  
}
```

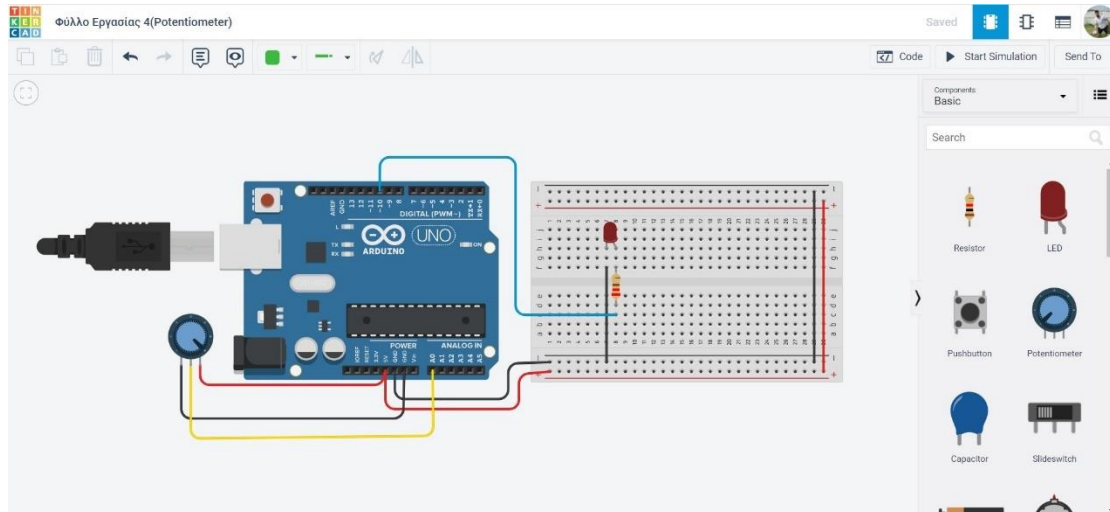
3.3.4 Φύλλο εργασίας 4 – Κλιμακωτή αύξηση και μείωση της φωτεινότητας με ποτενσιόμετρο

Στο συγκεκριμένο φύλλο εργασίας θα πραγματοποιήσουμε χρήση ενός ποτενσιόμετρου και με το εν λόγω εξάρτημα, θα ρυθμίζουμε την φωτεινότητα από το Led. Το ποτενσιόμετρο θεωρείται αναλογικό όργανο, κατά συνέπεια η σύνδεση του θα γίνει σε μια αναλογική είσοδο του Arduino, όπου θα διαβάζουμε τις τιμές. Η συνδεσμολογία από το Led και την αντίσταση θα είναι η ίδια με τα προηγούμενα κυκλώματα απλώς η έξοδος από το Led πρέπει να καταλήγει σε θύρα PWM.

54

Περιγραφή του κυκλώματος

Για το συγκεκριμένο φύλλο εργασίας θα κάνουμε χρήση ενός Led, καθώς και μίας αντίστασης όπου πρέπει να είναι συνδεδεμένα σε μια θύρα PWM. Επιπλέον θα πρέπει να κάνουμε την σύνδεση του ποτενσιόμετρου σε μία από τις έξι αναλογικές εισόδους (A0 – A5). Από εκεί θα παίρνουμε τις τιμές που αφορούν το ποτενσιόμετρο. Το κύκλωμα που αφορά το ποτενσιόμετρο θα παρατεθεί σε παράδειγμα, ωστόσο η συνδεσμολογία είναι: πραγματοποιούμε την σύνδεση της γείωσης και της τροφοδοσίας στα έξω ποδαράκια του ποτενσιόμετρου (Μπορούμε να τα βάλουμε από όποια πλευρά θέλουμε αυτό που θα αλλάξει είναι η φορά του διακόπτη μας). Στο μεσαίο ποδαράκι κάνουμε σύνδεση σε μία αναλογική είσοδο για τις τιμές του ποτενσιόμετρου. Οι τιμές που θα πάρει είναι από 0 έως 1023 κατά συνέπεια θα πρέπει να πραγματοποιήσουμε την συγκεκριμένη μετατροπή έτσι ώστε να δώσουμε την ένταση του Pin. Υπάρχει η δυνατότητα να συνδεθεί το ποτενσιόμετρο κατευθείαν πάνω στο Breadboard αρκεί να έχει τα κατάλληλα ποδαράκια.



Εικόνα 16- Παράδειγμα κυκλώματος με την σύνδεση ποτενσιόμετρου στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Στην συγκεκριμένη εφαρμογή θα πρέπει να δώσουμε προσοχή και να πραγματοποιήσουμε αρχικοποίηση όλων των θυρών. Ειδικά τα εξαρτήματα δεν θα λειτουργήσουν σωστά και δεν θα κάνουν αυτό που θέλουμε. Οπότε το Led το αρχικοποιούμε ως έξοδο, ενώ το ποτενσιόμετρο το αρχικοποιούμε ως είσοδο. Επιπλέον θα κάνουμε χρήση μίας μεταβλητής για να καταχωρούμε κάθε φορά την τιμή που θα παίρνουμε από το ποτενσιόμετρο. Επειδή όπως αναφέραμε και πριν οι τιμές θα είναι από το 0 έως το 1023, θα πρέπει να γίνει η προσαρμογή της εξόδου ως προς το Led και να διαιρεθεί με τον αριθμό 4. Η εντολή που θα χρησιμοποιήσουμε για την δουλειά αυτή είναι η AnalogWrite (Pin, Value) και η αμέσως επόμενη είναι η AnalogRead (Pin). Να υπενθυμίσουμε σ' αυτό το σημείο πως τα ονόματα από τα αναλογικά Pin είναι από το (A0 – A5) και στα ψηφιακά από το (0 – 13).

1. Πραγματοποιούμε τις αναγκαίες δηλώσεις που αφορούν τις εισόδους και τις εξόδους.
2. Θα πρέπει ξεχωριστά σε κάθε επανάληψη που υλοποιείται να διαβάζουμε την τιμή που μας δίνει το ποτενσιόμετρο και να την προσαρμόζουμε ως είσοδο στο Led.
3. Πραγματοποιούμε μια καθυστέρηση για να μπορούμε να βλέπουμε την διαφορά στην φωτεινότητα από το Led.

Οι έννοιες που μαθαίνω: Αναλογική είσοδος, Ποτενσιόμετρο

Ο κώδικας υλοποίησης



```
int ledPin = 10;

int potPin = A0;

int potValue = 0; // μεταβλητή για την τιμή που θα διαβάσω από το
ποτενσιόμετρο

void setup() {

  pinMode(ledPin, OUTPUT);

  pinMode(potPin, INPUT);

}

void loop() {

  potValue = analogRead(potPin);

  analogWrite(ledPin, potValue/4);

  delay(50);

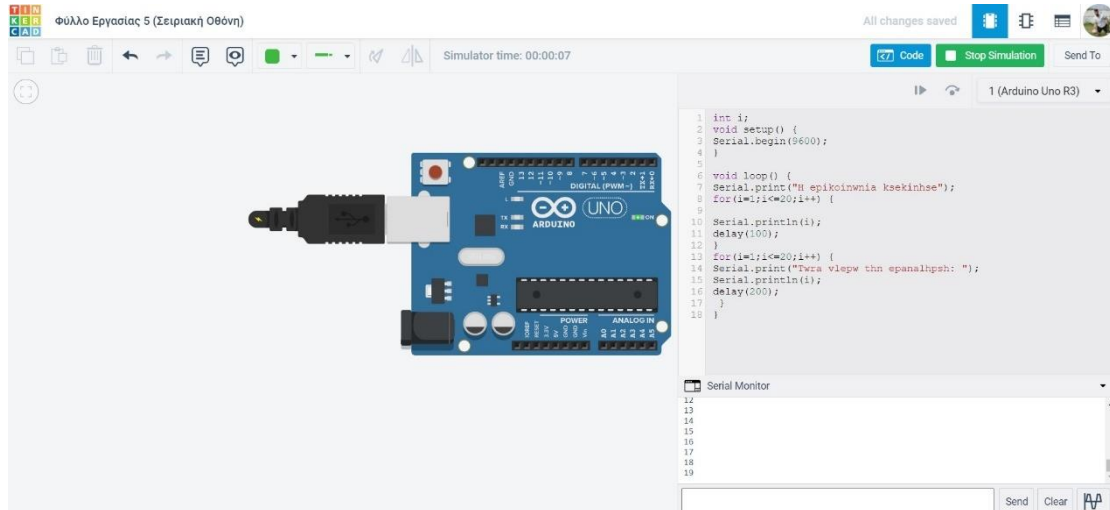
}
```

3.3.5 Φύλλο εργασίας 5 – Η χρήση της σειριακής οθόνης στο Arduino

Στην συγκεκριμένη εφαρμογή θα κάνουμε χρήση της σειριακής οθόνης του Arduino. Θα αποστείλουμε μηνύματα και θα λάβουμε τιμές μεταβλητών για να κατανοήσουμε πως λειτουργεί.

Περιγραφή του κυκλώματος

Στην εφαρμογή αυτή δεν θα χρειαστεί να υλοποιήσουμε κύκλωμα με κάποιο εξάρτημα, αυτό που πρέπει να κάνουμε για να είναι η οθόνη μας λειτουργική είναι απλά την σύνδεση του USB στο Arduino.



Εικόνα 17 - Παράδειγμα σειριακής οθόνης στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Στην εφαρμογή αυτή θα υλοποιήσουμε ένα πρόγραμμα το οποίο θα μας δίνει την δυνατότητα να ενεργοποιήσουμε την σειριακή θύρα και να εμφανίσουμε διάφορα μηνύματα στην οθόνη, με σκοπό να μάθουμε καλύτερα την σειριακή οθόνη.

1. Στη συνάρτηση setup() ενεργοποιούμε την σειριακή θύρα χρησιμοποιώντας ρυθμό δεδομένων 9600.
2. Σε κάθε επανάληψη της συνάρτησης loop() αποστέλλουμε το μήνυμα που θέλουμε να εμφανίσουμε χωρίς να πραγματοποιήσουμε αλλαγή γραμμής.
3. Κάνουμε χρήση μίας επαναληπτικής διαδικασίας για όσες επαναλήψεις εμείς θέλουμε, με σκοπό να εκτυπώσουμε τα μηνύματα που θέλουμε σε κάθε μία γραμμή.
4. Έχουμε την ευκαιρία να συντάξουμε ένα πιο σύνθετο μήνυμα το οποίο θα περιλαμβάνει και την μεταβλητή που χρησιμοποιούμε. Αυτό που εκτυπώνουμε έχουμε την δυνατότητα να το αλλάζουμε κάθε φορά γραμμή αν θέλουμε ή να μένει στην ίδια.
5. Ενεργοποιούμε την σειριακή οθόνη και παρακολουθούμε τα μηνύματα που λαμβάνουμε.
6. Προσοχή αν χρησιμοποιήσουμε ελληνικούς χαρακτήρες στο μήνυμα θα παρατηρήσουμε ότι δεν τυπώνονται σωστά οπότε τους αποφεύγουμε.

57

Παρατήρηση: Η χρήση της σειριακής οθόνης μπορεί να πραγματοποιηθεί για να βλέπουμε παράλληλα με την εφαρμογή μας τι συμβαίνει κατά της εκτέλεση του κώδικα.

Οι έννοιες που μαθαίνω: Σειριακή οθόνη.



Ο κώδικας υλοποίησης

```
int i;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print("Η επικοινωνία κσεκινήσ");  
  
  for(i=1;i<=20;i++) {  
    Serial.println(i);  
    delay(100);  
  }  
  
  for(i=1;i<=20;i++) {  
    Serial.print("Τωρα βλέπω την επανάληψη: ");  
    Serial.println(i);  
    delay(200);  
  }  
}
```

58

3.3.6 Φύλλο Εργασίας 6 – Χρησιμοποιώντας το RGB Led

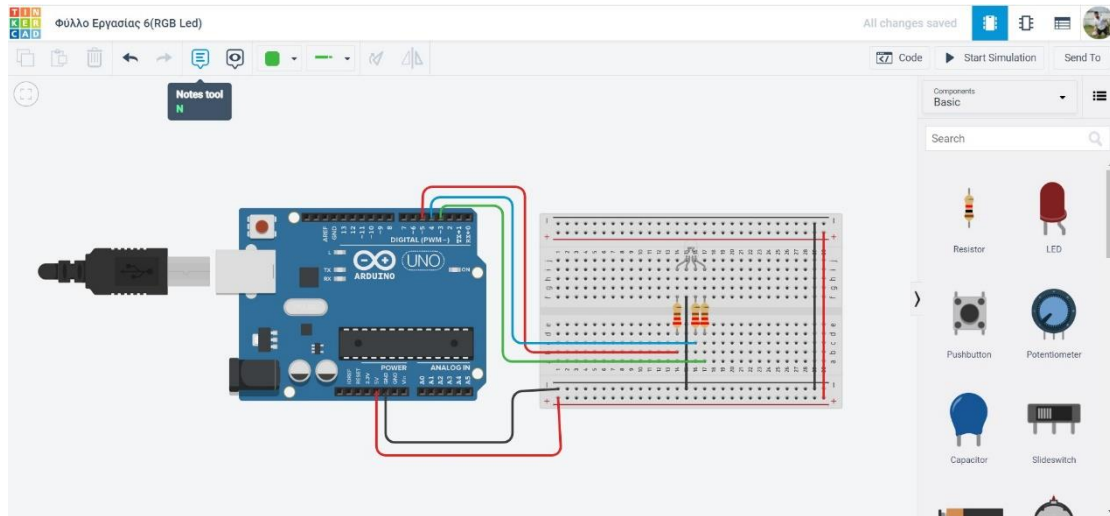
Στην εφαρμογή αυτή θα κάνουμε χρήση του RGB Led στο οποίο προγραμματίζοντας θα του δώσουμε την ευκαιρία να πάρει όλα τα χρώματα που μπορεί.

Περιγραφή του κυκλώματος

Γι' αυτή την εφαρμογή θα χρησιμοποιήσουμε ένα RGB Led και τρεις αντιστάσεις με τα ανάλογα Ω για κάθε ποδαράκι. Να υπενθυμίσουμε πως η τιμή που αφορά το ποδαράκι που αντιστοιχεί στο κόκκινο Led είναι λίγο μεγαλύτερη από τις άλλες δύο. Επιπλέον θα πρέπει να προσέξουμε την συνδεσμολογία pin και χρώματος. Το δεύτερο ποδαράκι από τ αριστερά το συνδέουμε στην γείωση, το πρώτο από τ αριστερά στο κόκκινο χρώμα, το τρίτο από τ αριστερά στο πράσινο χρώμα και το τέταρτο από τ



αριστερά στο μπλε. Να τονίσουμε πως όλες οι θύρες που θα χρησιμοποιήσουμε για το κάθε Led θα πρέπει να είναι PWM.



Εικόνα 18 - Παράδειγμα RGB Led στο Tinkercad

Ο προγραμματισμός του κυκλώματος

1. Πραγματοποιούμε τις αρχικοποιήσεις που είναι αναγκαίες και αφορούν τις θύρες, τις τιμές και τις μεταβλητές.
2. Υλοποιούμε μια διαδικασία επανάληψης και δίνουμε τιμές σε κάθε Led.
3. Πραγματοποιώντας μικρές αλλαγές στον κώδικα που θα παραθέσουμε θα καταφέρουμε να πετύχουμε όλους τους δυνατούς συνδυασμούς.

Οι έννοιες που μαθαίνω: RGB Led, Δομή επανάληψης.

Ο κώδικας υλοποίησης

```
int redPin = 5;  
int greenPin = 3;  
int bluePin = 4;  
int i;  
void setup() {  
pinMode(redPin, OUTPUT);
```



```
pinMode(greenPin, OUTPUT);  
pinMode(bluePin, OUTPUT);  
}  
void loop() {  
  // άναψε το κόκκινο  
  analogWrite(redPin, 255);  
  analogWrite(greenPin, 0);  
  analogWrite(bluePin, 0);  
  delay(2000);  
  // άναψε το πράσινο  
  analogWrite(redPin, 0);  
  analogWrite(greenPin, 255);  
  analogWrite(bluePin, 0);  
  delay(2000);  
  // άναψε το μπλε  
  analogWrite(redPin, 0);  
  analogWrite(greenPin, 0);  
  analogWrite(bluePin, 255);  
  delay(2000);  
}
```

Παρατήρηση: Κάνοντας χρήση μίας επανάληψης έχουμε την δυνατότητα να αλλάζουμε τις τιμές που χρησιμοποιούμε στα Led.

```
for (i =0; i<255; i=i+5) {  
  analogWrite(redPin, i);  
  analogWrite(greenPin, i);
```



```
analogWrite (bluePin, i);
```

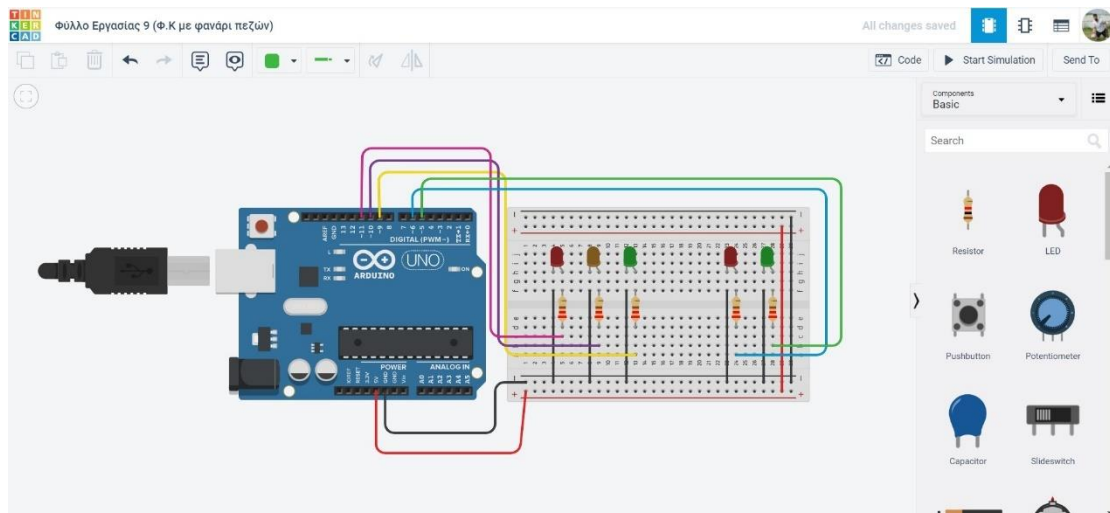
```
delay(200); }
```

3.3.7 Φύλλο εργασίας 7 – Τα φανάρια κυκλοφορίας μαζί με φανάρι πεζών με την βοήθεια των Led.

Στο συγκεκριμένο φύλλο εργασίας θα χρησιμοποιήσουμε 5 διαφορετικά Led με διαφορετικά χρώματα, για να προσομοιώσουμε ένα φανάρι κυκλοφορίας και ένα φανάρι πεζών. Έτσι με τον κατάλληλο προγραμματισμό θα καταφέρουμε όταν το φανάρι των πεζών είναι κόκκινο να είναι πράσινο το φανάρι της κυκλοφορίας, και μετά από κάποιο χρονικό διάστημα που εμείς θα επιλέξουμε το φανάρι της κυκλοφορίας θα γίνεται πορτοκαλί και στην συνέχεια κόκκινο και το φανάρι των πεζών θα καταλήγει πράσινο. Η διαδικασία θα επαναλαμβάνεται για όσο έχουμε το Arduino στο ρεύμα.

Περιγραφή του κυκλώματος

Γι' αυτό το φύλλο εργασίας θα χρειαστούμε 2 led κόκκινου χρώματος, 2 Led πράσινου χρώματος και 1 Led πορτοκαλί χρώματος. Επιπλέον θα χρειαστούμε 5 αντιστάσεις των 220 Ω για το κάθε Led. Παρακάτω θα παραθέσουμε το παράδειγμα του κυκλώματος. Η γείωση τοποθετείται ξεχωριστά σε κάθε ποδαράκι από τα Led.



Εικόνα 19 - Παράδειγμα Φαναριού κυκλοφορίας με φανάρι πεζών στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Ο προγραμματισμός θα βασιστεί στην περιγραφή που κάναμε παραπάνω. Αυτό που θα πρέπει να προσέξουμε είναι οι χρόνοι των φαναριών να συγχρονιστούν κατάλληλα έτσι ώστε τα δύο φανάρια ν' ανάβουν αντίστροφα.

Διπλωματική Εργασία



Ο κώδικας υλοποίησης

```
int ledRed = 11;

int ledOrange = 10;

int ledGreen = 9;

int pedRed = 6;

int pedGreen = 5;

void setup() {
  pinMode(ledRed, OUTPUT);
  pinMode(ledOrange, OUTPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(pedRed, OUTPUT);
  pinMode(pedGreen, OUTPUT);
}

void loop() {
  // κόκκινο για 3 δευτερόλεπτα, πράσινο στους πεζούς
  digitalWrite(ledRed, HIGH);
  digitalWrite(ledOrange, LOW);
  digitalWrite(ledGreen, LOW);
  digitalWrite(pedRed, LOW);
  digitalWrite(pedGreen, HIGH);
  delay(3000);

  // κόκκινο στους πεζούς, περιμένω για 1 δευτερόλεπτο πριν δώσω πράσινο στα
  αμάξια
  digitalWrite(pedRed, HIGH);
  digitalWrite(pedGreen, LOW);
  delay(1000);
```



```
// πράσινο κυκλοφορίας για 5 δευτερόλεπτα
```

```
digitalWrite(ledRed, LOW);
```

```
digitalWrite(ledOrange, LOW);
```

```
digitalWrite(ledGreen, HIGH);
```

```
delay(5000);
```

```
// πορτοκαλί για 1 δευτερόλεπτο
```

```
digitalWrite(ledRed, LOW);
```

```
digitalWrite(ledOrange, HIGH);
```

```
digitalWrite(ledGreen, LOW);
```

```
delay(1000);
```

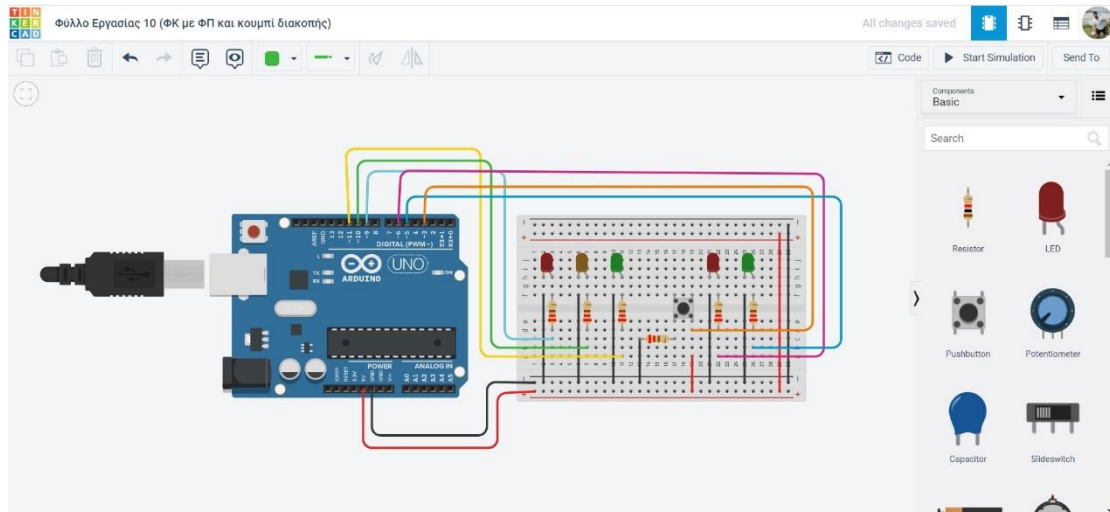
```
}
```

3.3.8 Φύλλο εργασίας 8 – Τα φανάρια κυκλοφορίας με το φανάρι πεζών και με button διακοπής.

Σ' αυτό το φύλλο εργασίας θα εμπλουτίσουμε το προηγούμενο φύλλο εργασίας προσθέτοντας ένα κουμπί το οποίο θα είναι υπεύθυνο για την διακοπή της κυκλοφορίας. Αρχικά θα είναι πράσινο το φανάρι της κυκλοφορίας και κόκκινο το φανάρι των πεζών και θα παραμένει. Όταν κάποιος θέλει να περάσει θα πατάει το κουμπί και το φανάρι της κυκλοφορίας, θα γίνεται σταδιακά πορτοκαλί και στην συνέχεια κόκκινο, ενώ το φανάρι των πεζών θα γίνεται πράσινο για συγκεκριμένη ώρα που εμείς έχουμε επιλέξει. Εν συνεχεία θα γίνεται κόκκινο το φανάρι των πεζών και η διαδικασία θα επαναλαμβάνεται από την αρχή μέχρις ότου απενεργοποιηθεί το Arduino.

Περιγραφή του κυκλώματος

Το κύκλωμα μας θα είναι ακριβώς ίδιο με το προηγούμενο φύλλο εργασίας, με την μόνη διαφορά πως θα χρειαστούμε ακόμη ένα κουμπί και μία αντίσταση της τάξεως των 220 Ω. Παρακάτω θα παραθέσουμε το παράδειγμα του κυκλώματος.



Εικόνα 20 - Παράδειγμα φαναριού κυκλοφορίας με φανάρι πεζών και κουμπί διακοπής στο Tinkercad

Ο προγραμματισμός του κυκλώματος

1. Πραγματοποιούμε τις αναγκαίες αρχικοποιήσεις για τις μεταβλητές, τις τιμές, την είσοδο και την έξοδο.
2. Ρυθμίζουμε το χρονικό διάστημα για το οποίο θέλουμε να εναλλάσσονται τα φανάρια της κυκλοφορίας και των πεζών.
3. Διαβάζουμε την είσοδο που αφορά το Pin του κουμπιού και όταν αυτή η είσοδος αναγνωρίζεται, υλοποιούμε τα παραπάνω βήματα. Χωρίς να είναι απαραίτητο αλλά θα ήταν αρκετά βοηθητικό να χρησιμοποιήσουμε την συνάρτηση millis(). Με την συγκεκριμένη συνάρτηση θα έχουμε την δυνατότητα της καταγραφής του χρόνου για να γνωρίζουμε πότε έχει πατηθεί το κουμπί. Επιπλέον θέτουμε ένα χρονικό διάστημα το οποίο δεν θα ανάβει το πράσινο φανάρι των πεζών.

64

Προσοχή: Θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί για να ξεχωρίσουμε ποια βήματα γίνονται μόνο για μία φορά και ποια γίνονται σε επανάληψη.

Οι έννοιες που μαθαίνω: Κουμπιά (Button), Καταγραφή χρόνου

Ο κώδικας υλοποίησης

```
int ledRed = 9;  
int ledOrange = 10;  
int ledGreen = 11;
```




```
int pedRed = 6;

int pedGreen = 5;

int btn = 3; //συνδέω το κουμπί στο pin 5

int lastPedPass = 0; //πότε διακόπηκε τελευταία φορά η κυκλοφορία για να
περάσει πεζός

int btnState; //για να διαβάζουμε αν πατήθηκε το κουμπί μας

void setup() {
  pinMode(ledRed, OUTPUT);
  pinMode(ledOrange, OUTPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(pedRed, OUTPUT);
  pinMode(pedGreen, OUTPUT);
  pinMode(btn, INPUT);

  // κόκκινο στους πεζούς και πράσινο στα αυτοκίνητα αρχικά
  digitalWrite(pedRed, HIGH);
  digitalWrite(pedGreen, LOW);
  digitalWrite(ledRed, LOW);
  digitalWrite(ledOrange, LOW);
  digitalWrite(ledGreen, HIGH);
}

void loop() {
  int btnState = digitalRead(btn);
  if (btnState == HIGH && (millis() - lastPedPass) > 5000) {
```



```
// αν πατήθηκε το κουμπί και ο χρόνος από την τελευταία αλλαγή είναι πάνω από  
5 sec  
  
// πορτοκαλί για 1 δευτερόλεπτο  
digitalWrite(ledRed, LOW);  
digitalWrite(ledOrange, HIGH);  
digitalWrite(ledGreen, LOW);  
delay(1000);  
  
// κόκκινο στα αμάξια για 5 δευτερόλεπτα, περιμένω 1 sec και δίνω πράσινο  
στους πεζούς  
digitalWrite(ledRed, HIGH);  
digitalWrite(ledOrange, LOW);  
digitalWrite(ledGreen, LOW);  
delay(1000);  
digitalWrite(pedRed, LOW);  
digitalWrite(pedGreen, HIGH);  
delay(3000);  
  
// κόκκινο στους πεζούς, περιμένω για 1 δευτερόλεπτο πριν δώσω πράσινο στα  
αμάξια  
digitalWrite(pedRed, HIGH);  
digitalWrite(pedGreen, LOW);  
delay(1000);  
  
// πράσινο κυκλοφορίας για 5 δευτερόλεπτα  
digitalWrite(ledRed, LOW);  
digitalWrite(ledOrange, LOW);  
digitalWrite(ledGreen, HIGH);  
lastPedPass = millis(); // κρατάω το χρόνο της τελευταίας αλλαγής φαναριών  
}}
```

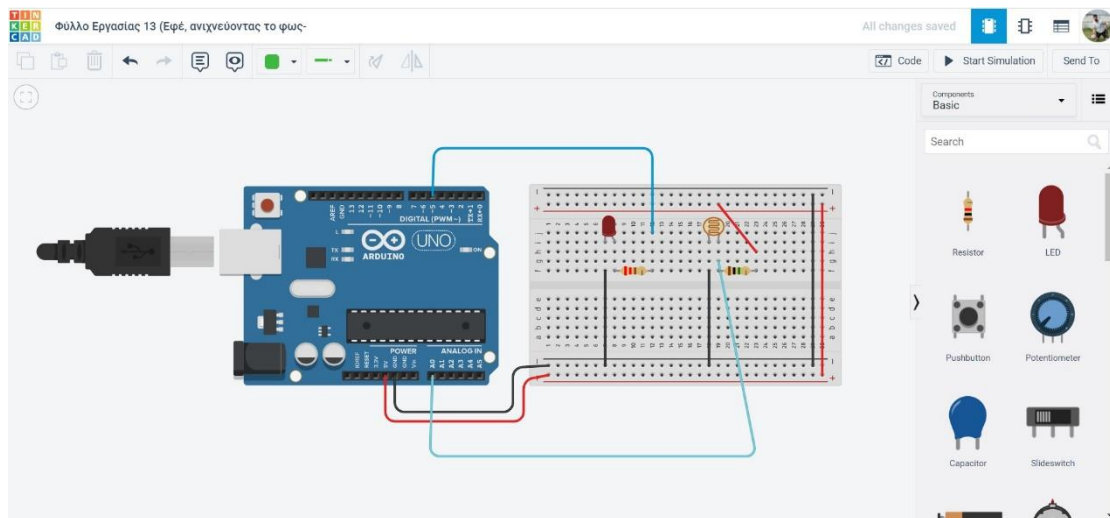


3.3.9 Φύλλο Εργασίας 9 – Ανιχνεύοντας το φως με την βοήθεια φωτοαντίστασης

Στην συγκεκριμένη εφαρμογή θα υλοποιήσουμε ένα κύκλωμα με ένα Led και μια φωτοαντίσταση. Κατά συνέπεια το Led θα ανταποκρίνεται στα ερεθίσματα που παίρνει η φωτοαντίσταση από το φως που λαμβάνει.

Περιγραφή του κυκλώματος

Για το φύλλο εργασίας αυτό θα κάνουμε χρήση ενός Led, μίας φωτοαντίστασης και δύο αντιστάσεων οι οποίες η μία θα είναι συνδεδεμένη με το Led και η άλλη με την φωτοαντίσταση. Θα κάνουμε χρήση της φωτοαντίστασης για να λαμβάνουμε την φωτεινότητα του χώρου. Οπότε ανάλογα με την φωτεινότητα του χώρου θα ρυθμίζεται η φωτεινότητα του Led. Η αντίσταση που συνδέεται με το Led είναι 220 Ω, ενώ η αντίσταση που συνδέεται με την φωτοαντίσταση είναι περίπου 1 ΜΩ. Θα παρατεθεί παρακάτω το κύκλωμα με τα εξαρτήματα. Η διάταξη του κυκλώματος μοιάζει αρκετά με το φύλλο εργασίας του ποτενσιόμετρου, με την μόνη διαφορά πως το ρόλο του διακόπτη του παίζει η φωτοαντίσταση.



Εικόνα 21 - Παράδειγμα κυκλώματος με την βοήθεια φωτοαντίστασης στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Όπως στην υλοποίηση του κυκλώματος έτσι και στον προγραμματισμό η φιλοσοφία που ακολουθούμε είναι εντελώς ίδια με αυτή του ποτενσιόμετρου. Χρησιμοποιώντας την αναλογική είσοδο διαβάζουμε την τιμή που μας ενδιαφέρει και γίνεται προσαρμογή της συγκεκριμένης τιμής, με σκοπό να πάρουμε την αναλογική έξοδο που μας ενδιαφέρει για το Led μας. Να τονίσουμε πως θα πρέπει να γίνει



προσαρμογή της εισόδου μας καθώς το εύρος των τιμών που έχει είναι από 0 έως 1023, ως προς την έξοδο του Led μας που οι τιμές του είναι από 0 έως 255. Αυτή η διαδικασία υλοποιείται είτε με την συνάρτηση map την οποία έχουμε δει πως συντάσσετε, ή απλά διαιρούμε την τιμή εισόδου με τον αριθμό 4.

Εφικτές δοκιμές στο κύκλωμα: Με το συγκεκριμένο κύκλωμα έχουμε την ευκαιρία να πειραματιστούμε. Κατά συνέπεια θα είχαμε την δυνατότητα να κάνουμε σύνδεση του ενός ποδιού στην πηγή και το άλλο πόδι της κανονικής αντίστασης στην γείωση. Έπειτα μπορούμε να πραγματοποιήσουμε ακριβώς το αντίθετο για να δούμε τι θα συμβεί. Ακόμη μία ενέργεια που έχουμε την δυνατότητα να κάνουμε για να δούμε τι θα συμβεί είναι, να πραγματοποιήσουμε την σύνδεση της φωτοαντίστασης με καλώδια και να κινηθούμε στον χώρο για να διακρίνουμε τις αλλαγές της φωτεινότητας που θα έχουμε.

Οι έννοιες που μαθαίνω: Η συνάρτηση αντιστοίχισης τιμών map, Φωτοαντίσταση.

Ο κώδικας υλοποίησης:

```
int ledPin = 5;
int resPin = A0;
int resVal = 0; // εδώ θα διαβάζουμε την τιμή της αντίστασης
int ledVal = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(resPin, INPUT);
}
void loop() {
  resVal = analogRead(resPin);
  ledVal = map(resVal, 0, 1023, 0, 255);
  analogWrite(ledPin, ledVal);
  // αν δεν χρησιμοποιούσα την map, θα έγραφα analogWrite(ledPin, resVal / 4);
  delay(50);
}
```

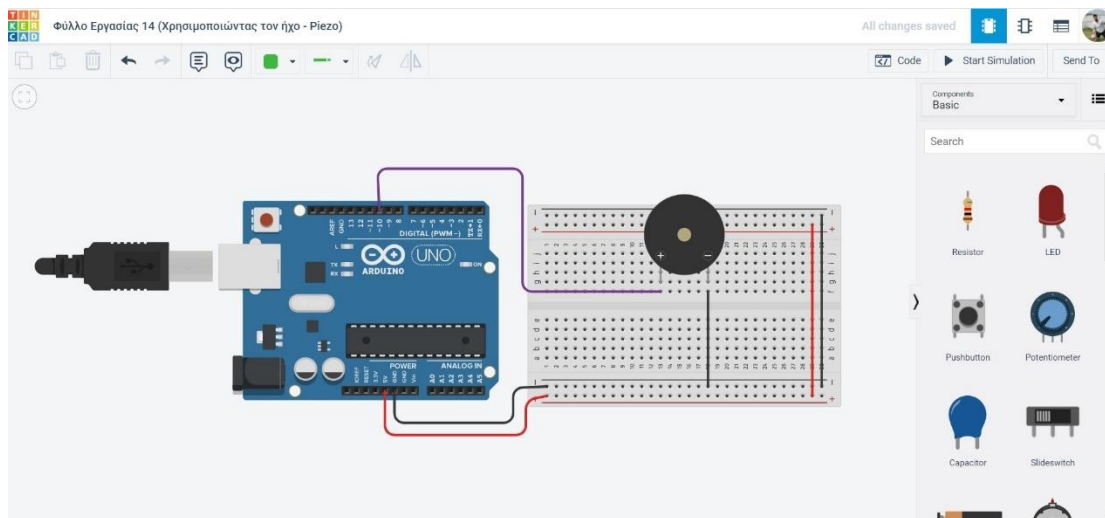


3.3.10 Φύλλο εργασίας 10 – Χρησιμοποιώντας τον ήχο με Piezo

Στη συγκεκριμένη εφαρμογή θα κάνουμε χρήση ενός απλού κυκλώματος με την βοήθεια ενός ηχείου (Piezo). Αρκετά από τα εξαρτήματα που χρησιμοποιούμε στο Arduino μόλις δεχθούν συνεχές ρεύμα παράγουν ήχο. Υπάρχουν διάφοροι τύποι ηχείων ωστόσο εμείς θα δούμε το Piezo. Οπότε σ' αυτό το φύλλο εργασίας υλοποιώντας αυξομείωση στην τιμή του ρεύματος θα παράγουμε διάφορους ήχους.

Περιγραφή του κυκλώματος

Σ' αυτό το κύκλωμα θα κάνουμε χρήση ενός Piezo. Τα Piezo λειτουργούν με συνεχή τάση χρησιμοποιώντας την τροφοδοσία και την γείωση σε δύο ακροδέκτες. Η τιμή της τροφοδοσίας είναι στα 5V ενώ ο άλλος ακροδέκτης συνδέεται στην γείωση. Αν η τάση που παρέχουμε στο Piezo είναι σταθερή τότε και ο ήχος θα σταθερός. Κατά συνέπεια η έξοδος που θα χρησιμοποιήσουμε από το Arduino θα είναι PWM, έτσι ώστε να έχουμε μία ψευδοαναλογική έξοδο και να πάρουμε όλο το εύρος των τιμών με την αυξομείωση.



Εικόνα 22 - Παράδειγμα κυκλώματος Piezo στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Η υλοποίηση του συγκεκριμένου κυκλώματος είναι αρκετά εύκολη. Οπότε αυτό που κάνουμε είναι να αρχικοποιήσουμε το αναγκαίο Pin στο Arduino και δίνοντας τις κατάλληλες εντολές, να λάβουμε τους διαφορετικούς ήχους.

1. Υλοποίηση αρχικοποιήσεων
2. Χρησιμοποιούμε μια επαναληπτική διαδικασία με σκοπό να δώσουμε σταδιακά τιμές στο Pin μας, λαμβάνοντας τους ήχους που θέλουμε. Το εύρος



των τιμών που θα δώσουμε είναι από το 0 έως το 255 για ν' ακούσουμε όλους τους ενδιάμεσους ήχους, δίνοντας βέβαια ενδιάμεσο χρόνο ειδικά αυτό δεν θα επιτευχθεί.

3. Η διαδικασία αυτή μπορεί να υλοποιηθεί με τον ακριβώς αντίστροφο τρόπο ξεκινώντας από το 255 έως το 0.

Οι έννοιες που μαθαίνω: Ηχείο (Piezo)

Ο κώδικας υλοποίησης

```
int sndPin = 10;
int i;
void setup() {
  pinMode(sndPin, OUTPUT);
}
void loop() {
  for (i=0;i<=255;i=i+5) {
    analogWrite(sndPin, i);
    delay(50);
  }
}
```

70

3.3.11 Φύλλο εργασίας 11 – Ήχος με Piezo και αντίστροφή φωτός με φωτοαντίσταση

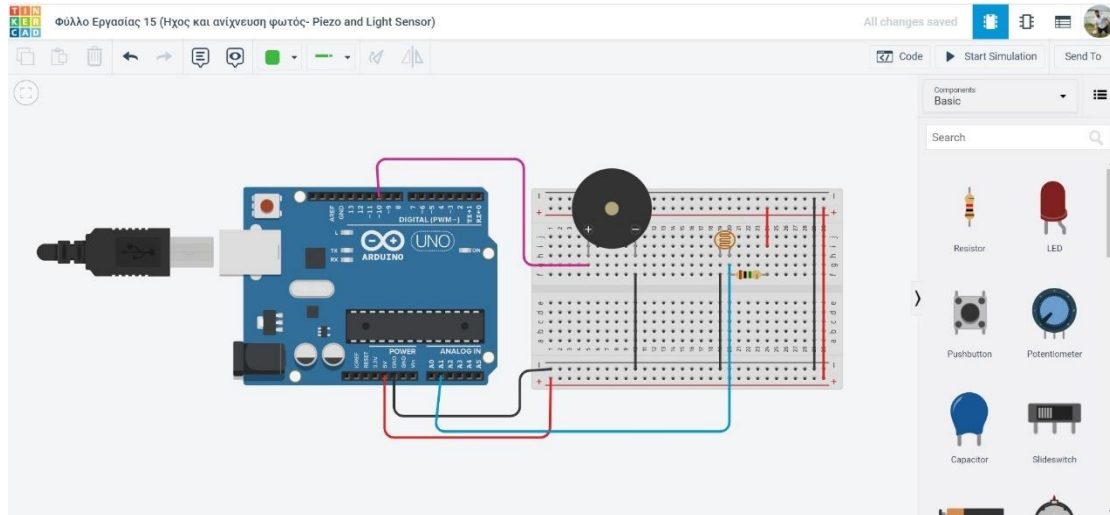
Η συγκεκριμένη εφαρμογή είναι άμεσα συνδεδεμένη με τα δύο προηγούμενα φύλλα εργασίας. Θα λειτουργεί με την βοήθεια μίας φωτοαντίστασης η οποία αλλάζοντας την τιμή της θα ρυθμίζει τον ήχο σε ένα Piezo.

Περιγραφή του κυκλώματος

Γι' αυτό το φύλλο εργασίας τα αναγκαία εξαρτήματα είναι ένα Piezo, μια φωτοαντίσταση και μία αντίσταση της τάξεως του 1ΜΩ περίπου. Θα χρησιμοποιήσουμε την φωτοαντίσταση με τον ίδιο τρόπο που σε προηγούμενο φύλλο



είχαμε χρησιμοποιήσει το ποτενσιόμετρο. Έτσι η φωτοαντίσταση θα παίζει τον ρόλο του ρυθμιστή του ήχου ανάλογα με το φως το οποίο λαμβάνει. Παρακάτω θα παραθέσουμε το κύκλωμα με την ανάλογη συνδεσμολογία.



Εικόνα 23 - Παράδειγμα κυκλώματος Piezo με την χρήση φωτοαντίστασης στο Tinkercad

Ο προγραμματισμός του κυκλώματος

Ο προγραμματισμός που θ' ακολουθήσουμε είναι κοινός με το φύλλο εργασίας του ποτενσιόμετρου. Κατά συνέπεια η διαδικασία είναι η εξής: παίρνουμε μία τιμή από μία αναλογική θύρα του Arduino ως είσοδο και με τους τρόπους που είδαμε παραπάνω, την μορφοποιούμε από 0 – 1023 σε 0 – 255 στην έξοδο μας.

Εφικτές δοκιμές στο κύκλωμα: Έχουμε την δυνατότητα να πραγματοποιήσουμε ακριβώς τις ίδιες δοκιμές στο κύκλωμα μας με το παραπάνω φύλλο εργασίας που αφορά την φωτοαντίσταση.

Ο κώδικας υλοποίησης

```
int sndPin = 10;  
int resPin = A1;  
int resVal = 0; // εδώ θα διαβάζουμε την τιμή της αντίστασης  
int ledVal = 0;  
void setup() {  
  pinMode(sndPin, OUTPUT);
```



```
pinMode(resPin, INPUT);  
}  
void loop() {  
  resVal = analogRead(resPin);  
  ledVal = map(resVal, 0, 1023, 0, 255);  
  analogWrite(sndPin, ledVal);  
  // αν δεν χρησιμοποιούσα την map, θα έγραφα analogWrite(sndPin, resVal / 4);  
  delay(50);  
}
```

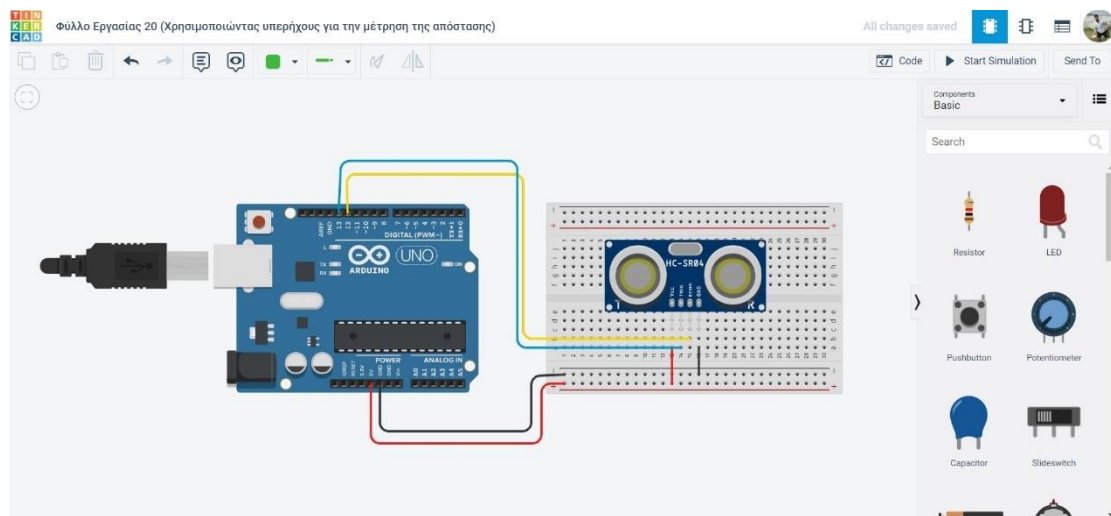
3.3.12 Φύλλο εργασίας 12 – Χρησιμοποιώντας τον αισθητήρα για την μέτρηση απόστασης (HC-SR04)

Στην τελευταία μας εφαρμογή θα υλοποιήσουμε ένα κύκλωμα με το εξάρτημα υπερήχων (HC – SR04) που θα το χρησιμοποιήσουμε για την μέτρηση της απόστασης. Έχουμε αναφέρει σχετικές πληροφορίες για το εξάρτημα στην θεωρία που έχουμε παρουσιάσει πιο πάνω.

Περιγραφή του κυκλώματος

Το συγκεκριμένο εξάρτημα περιλαμβάνει 4 υποδοχές διασύνδεσης. Στο πρώτο ποδαράκι (διάταξη από αριστερά προς δεξιά) συνδέουμε την τροφοδοσία των 5V, στο δεύτερο από τ αριστερά το trigger το οποίο είναι υπεύθυνο για την αποστολή του σήματος, στο αμέσως επόμενο το echo που είναι υπεύθυνο για την λήψη κάποιου σήματος και στο τέταρτο συνδέουμε την γείωση.

72



Εικόνα 24 - Παράδειγμα κυκλώματος με αισθητήρα υπερήχων (HC-SR04) στο Tinkercad



Ο προγραμματισμός του κυκλώματος

Υλοποιούμε την κατάλληλη συνδεσμολογία των ακροδεκτών στο Arduino. Είναι απαραίτητο όταν προγραμματίσουμε να αποστέλλουμε ένα σήμα, περιμένοντας ένα μικρό χρονικό διάστημα για να το διαβάσουμε. Έπειτα ακολουθούμε τα εξής βήματα:

1. Ορίζουμε τα Pin θέλουμε ως είσοδο ή έξοδο.
2. Είναι αναγκαίο να ενεργοποιήσουμε την σειριακή οθόνη με τον γνωστό μας ρυθμό 9600.
3. Θέτουμε δύο όρια σε δύο τιμές οι οποίες θα διαβάζονται σε εκατοστά, (π.χ οι τιμές είναι 0 και 200) για να υπάρχει έλεγχος όπου οι τιμές που διαβάζονται είναι από 0 έως 2 μέτρα.
4. Για να βεβαιωθούμε πως δεν αποστέλλετε κάτι μηδενίζουμε τον παλμό του trigger, κάνοντας χρήση της συνάρτησης digitalWrite () και αναμένουμε για 2 μς.
5. Αποστέλλουμε παλμό στο trigger και αναμένουμε για 10 μς.
6. Έπειτα είναι αναγκαίο να μηδενίσουμε τον παλμό, και με την βοήθεια της συνάρτησης pulseIn(echoPin, HIGH) διαβάζουμε από το echo πόσο θα διαρκέσει ο παλμός.
7. Αφότου υλοποιηθεί το προηγούμενο βήμα ακολουθεί ο υπολογισμός της απόστασης που αφορά το εμπόδιο το οποίο έχουμε μπροστά μας από τον τύπο $distance = duration/58.2$.
8. Πραγματοποιούμε έλεγχο αν η απόσταση βρίσκεται στα επιθυμητά όρια.
9. Αποστέλλουμε την απόσταση στην σειριακή οθόνη σε περίπτωση που θέλουμε να πραγματοποιήσουμε διάφορες δοκιμές.
10. Τοποθετούμε μια μικρή καθυστέρηση της τάξεως των 50ms, έτσι ώστε το πρόγραμμα μας να ξεκινήσει ξανά τον έλεγχο.

Οι έννοιες που μαθαίνω: Αισθητήρας υπερήχων (HC-SR04), καθυστέρηση.

Ο κώδικας υλοποίησης

```
int echoPin = 12; // Echo Pin
int trigPin = 13; // Trigger Pin
int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
```



```
long duration, distance; // Duration used to calculate distance
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  /* Στέλνουμε κύματα ήχου και τα παίρνουμε πίσω, προσπαθώντας έτσι να
  μετρήσουμε την
  απόσταση από την ταχύτητα που έχουν. */
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  //Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
  distance = duration/58.2;
  if (distance >= maximumRange || distance <= minimumRange){
    /* εκτός ορίων */
    Serial.println("Εκτός ορίων");
    delay(100);
  }
  else {
    /* επιτυχής ανάγνωση */
    Serial.println(distance);
    delay(50);
  };
  //Καθυστέρηση 50ms πριν την επόμενη ανάγνωση
  delay(50);
}
```



3.4 Διδακτικά σενάρια με την πλατφόρμα του Arduino

Στο παρακάτω υποκεφάλαιο θα υλοποιηθούν δύο διδακτικά σενάρια, τα οποία θα είναι βασισμένα σε όλα τα εξαρτήματα τα οποία χρησιμοποιήσαμε στον παραπάνω οδηγό εκμάθησης για αρχάριους στο Arduino. Η δυσκολία των σεναρίων κατατάσσεται στην Γ τάξη του γενικού λυκείου. Οι εκπαιδευόμενοι θα πρέπει να έχουν εκπονήσει τον παραπάνω οδηγό για να μπορέσουν να τα φέρουν εις πέρας.

3.4.1 Το πρώτο διδακτικό σενάριο

Παιχνίδι μνήμης με Arduino και προσομοίωση στο Tinkercad στη Γ Λυκείου

1.1 Τίτλος εκπαιδευτικού σεναρίου

Παιχνίδι μνήμης με Arduino και προσομοίωση στο Tinkercad.

2.1 Εκτιμώμενη διάρκεια εκπαιδευτικού σεναρίου

Για να πραγματοποιηθεί το συγκεκριμένο σενάριο θα χρειαστούμε 6 διδακτικές ώρες. Ο εκπαιδευτικός θα πρέπει να κάνει τον διαχωρισμό σε 2 ώρες θεωρία και σε 4 ώρες στην υλοποίηση των ασκήσεων.

3.1 Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών – Απαιτούμενες γνώσεις

Το εν λόγω σενάριο απευθύνεται σε μαθητές της Γ Λυκείου και θα πρέπει να έχουν κάποιες απαιτούμενες γνώσεις με σκοπό να το εκπονήσουν. Το συγκεκριμένο εκπαιδευτικό σενάριο ακολουθεί το αναλυτικό πρόγραμμα σπουδών της Γ Λυκείου και το μάθημα της πληροφορικής και εντάσσεται προς το τέλος του κεφαλαίου του δομημένου προγραμματισμού. Οι μαθητές θα πρέπει να είναι σε θέση να χειριστούν απλούς αλγορίθμους καθώς και να μπορούν να συντάξουν βασικές εντολές στο δομημένο προγραμματισμό. Επίσης θα ήταν καλό να υπάρχει καλή γνώση Η/Υ διότι θα χειριστούν το Arduino και την πλατφόρμα του Tinkercad.



4.1 Σκοποί και στόχοι του εκπαιδευτικού σεναρίου

Ο σκοπός του εκπαιδευτικού σεναρίου είναι οι μαθητές να δημιουργήσουν ένα Παιχνίδι μνήμης πρώτα στο Tinkercad και έπειτα αφού είναι λειτουργικό να μεταφερθεί στην πλακέτα του Arduino, δοκιμάζοντας διάφορες τεχνικές επίλυσης με εντολές δομημένου προγραμματισμού.

Στόχοι εκπαιδευτικού σεναρίου:

Με την ολοκλήρωση του εκπαιδευτικού σεναρίου οι μαθητές θα πρέπει:

- ✚ Να δοκιμάζουν διαφορετικές τεχνικές επίλυσης ενός προβλήματος.
- ✚ Να μάθουν να δημιουργούν έξυπνες κατασκευές μέσω της πλατφόρμας του Tinkercad αλλά και του Arduino.
- ✚ Να καταλάβουν την έννοια την ομαδοσυνεργατικότητα.
- ✚ Να έχουν την δυνατότητα να συνδυάσουν το παιχνίδι με την μάθηση.
- ✚ Να κωδικοποιούν αλγορίθμους σε διαφορετικά προγραμματιστικά περιβάλλοντα (Arduino, Tinkercad).

5.1 Περιγραφή εκπαιδευτικού σεναρίου

Οι μαθητές χρησιμοποιώντας διάφορες τεχνικές προγραμματισμού θα δημιουργήσουν ένα παιχνίδι μνήμης. Η δημιουργία του παιχνιδιού θα βοηθήσει τους μαθητές να κατανοήσουν την σύνταξη αλλά και το πως λειτουργεί ο δομημένος προγραμματισμός, ενώ παράλληλα εξασκούν και την μνήμη τους. Επιπλέον μαθαίνουν να φτιάχνουν απλές κατασκευές στην πλατφόρμα του Tinkercad αλλά και στο Arduino. Επειδή οι μαθητές εργάζονται σε ομάδες καλλιεργούν το αίσθημα της ευθύνης και την υπευθυνότητας και επιπλέον ανταλλάζουν ιδέες και απόψεις. Με τον τρόπο αυτό οδηγούνται πιο εύκολα στην λύση της άσκησης.

Στο συγκεκριμένο σενάριο θα χρησιμοποιηθούν διάφορες διδακτικές προσεγγίσεις, οι οποίες θα βοηθήσουν τους μαθητές να κατανοήσουν το μάθημα. Επιπρόσθετα θα μάθουν να προσαρμόζονται ευκολότερα σε νέους τρόπους εκπαίδευσης. Οι προσεγγίσεις που θα χρησιμοποιήσουμε είναι οι εξής:

- ✚ **Θεωρητικό μέρος:** Στο μέρος αυτό αρχίζοντας το σενάριο οι μαθητές θα υλοποιήσουν μια επανάληψη στον δομημένο προγραμματισμό, στο πως λειτουργεί και στην σύνταξή του. Επίσης θα γνωρίσουν πως χρησιμοποιείται το λογισμικό αλλά και η πλακέτα Arduino και θα γίνει μια υπενθύμιση στην πλατφόρμα του Tinkercad.



- Πρακτικό μέρος:** Αρχικά οι μαθητές ξεκινάν και λύνουν το φύλλο εργασίας, εφαρμόζοντας όσα έχουν διδαχθεί στο θεωρητικό μέρος. Εν συνεχεία οι μαθητές θα μάθουν να χειρίζονται το λογισμικό του Arduino λύνοντας διάφορους αλγορίθμους. Ενώ ταυτόχρονα με τον τρόπο αυτό βελτιώνουν τις γνώσεις τους στο δομημένο προγραμματισμό.
- Ομαδικό μέρος:** Στο μέρος αυτό οι μαθητές θα πρέπει να συνεργαστούν για να εκπονήσουν το εκπαιδευτικό σενάριο, με συνέπεια ν' αναπτύξουν την έννοια της ομαδοσυνεργατικότητας. Ο κάθε μαθητής θ αναλάβει το δικό του σκέλος στην άσκηση με αποτέλεσμα να τον κάνει περισσότερο υπεύθυνο. Επίσης θα αποκτήσουν μεγαλύτερο ενδιαφέρον για το μάθημα διότι θα ζυμωθούν με τους συμμαθητές τους και θ' ανταλλάξουν απόψεις καθώς και γνώσεις πάνω στο αντικείμενο.
- Εξερευνητικό μέρος:** Στο συγκεκριμένο μέρος οι μαθητές θα εξερευνήσουν πιθανές λύσεις του προβλήματος πάνω στην δραστηριότητα που τους ανατέθηκε. Θα δοκιμάσουν διάφορες τεχνικές επίλυσης μέχρι να οδηγηθούν στην λύση.

6.1 Επιστημολογική προσέγγιση και εννοιολογική ανάλυση – θέματα θεωρίας του εκπαιδευτικού σεναρίου

Στο εν λόγω εκπαιδευτικό σενάριο έχει εφαρμοστεί η μέθοδος του Extreme Programming που πρώτος την χρησιμοποίησε ο Kent Beck. Η συγκεκριμένη μέθοδος είναι βασισμένη στον προγραμματισμό ανά ζεύγη με σκοπό να υπάρχει πλήρης έλεγχος στον κώδικα. Ο στόχος της μεθόδου αυτής είναι η μεγιστοποίηση της παραγωγικότητας των μαθητών αλλά και η δημιουργία υψηλότερου επιπέδου προγραμμάτων. Για να μπορέσει να επιτευχθεί αυτή η μέθοδος θα πρέπει να κάνουμε πράξη τέσσερις βασικές λειτουργίες της οι οποίες είναι: ο προγραμματισμός, ο έλεγχος, η κατανόηση των αναγκών του χρήστη και ο σχεδιασμός της εμφάνισης του προγράμματος.

7.1 Χρήση Η/Υ και ψηφιακών μέσω για το εκπαιδευτικό σενάριο

Για να μπορέσουν οι μαθητές να επιλύσουν το συγκεκριμένο φύλλο εργασίας, θα πρέπει να έχουν στον Η/Υ εγκαταστημένο το λογισμικό του Arduino και να έχουν πρόσβαση στο διαδίκτυο καθώς θα εργαστούν και στο Tinkercad. Επίσης θα πρέπει κάθε ομάδα να έχει την δική της ξεχωριστή πλακέτα.



8.1 Πρόβλεψη δυσκολιών μαθητών στο εκπαιδευτικό σενάριο

Αρκετές φορές χρησιμοποιώντας μια νέα μέθοδο διδασκαλίας μπορεί να ελλοχεύει δυσκολίες στην εκμάθηση ενός νέου σεναρίου για τους μαθητές. Ειδικότερα οι μαθητές δυσκολεύονται στο να δουλέψουν σε κάτι καινούργιο όπως το να κατασκευάσουν μικρά κυκλώματα ή ένα καινούργιο προγραμματιστικό περιβάλλον. Έτσι λοιπόν δουλεύοντας με παραδείγματα και διάφορες ασκήσεις πάνω σ' αυτά τους βοηθά να προσαρμοστούν πιο γρήγορα και να τα επιλύσουν. Η χρήση του λογισμικού Arduino καθώς και η πλατφόρμα του Tinkercad είναι πολύ εύκολα στην χρήση άσχετα που και τα δύο είναι στην αγγλική γλώσσα. Οι λέξεις που περιλαμβάνει για την Γ Λυκείου είναι απλές για τους μαθητές. Επίσης η εγκατάσταση του προγράμματος δεν χρειάζεται ιδιαίτερους πόρους συστήματος καθώς είναι συμβατό με τα περισσότερα λειτουργικά και δεν έχει ιδιαίτερες απαιτήσεις.

9.1 Διδακτικό συμβόλαιο

Στη διάρκεια του εκπαιδευτικού σεναρίου ο εκπαιδευτικός θα πρέπει να έχει καθοδηγητικό ρόλο απέναντι στους μαθητές. Αυτό θα έχει σαν συνέπεια να επικρατήσει το φιλικό κλίμα στην τάξη και το ομαδικό. Επίσης είναι σημαντικό ο εκπαιδευτικός να βοηθά και να εμπνέει τους μαθητές με σκοπό να φέρουν εις πέρας της άσκηση που τους έχει ανατεθεί.

10.1 Υποκείμενη θεωρία μάθησης

Οι μαθητές θέτουν ως στόχο την επίλυση του φύλλου εργασίας με διάφορα στοιχεία του δομημένου προγραμματισμού ενώ είναι χωρισμένοι σε ομάδες. Το φύλλο εργασίας που θα τους ανατεθεί και η διαδικασία επίλυσης του θα μας βοηθήσουν σ' αυτό το σκοπό. Κατασκευάζοντας οι μαθητές το συγκεκριμένο παιχνίδι μνήμης κάνουν το μάθημα διασκεδαστικότερο ενώ ταυτόχρονα κατανοούν διάφορες λειτουργίες και τεχνικές επίλυσης του δομημένου προγραμματισμού. Οι μαθητές είναι χωρισμένοι σε ομάδες ανάλογα με το γνωστικό επίπεδο που βρίσκονται και παράλληλα γίνεται ανταλλαγή ιδεών, με σκοπό την επίλυση του φύλλου εργασίας. Ο σκοπός είναι η δημιουργία ευχάριστου κλίματος στην τάξη για να κάνει το μάθημα επιτυχή.



11.1 Οργάνωση της τάξης

Για την πραγματοποίηση του εκπαιδευτικού σεναρίου οι μαθητές θα χωριστούν σε ομάδες των δύο ατόμων. Ο συγκεκριμένος διαχωρισμός γίνεται από τον εκπαιδευτικό για να δημιουργήσει ισορροπία μεταξύ των ομάδων και δίνει βάση στο γνωστικό επίπεδο του κάθε μαθητή. Οι μαθητές θα πρέπει να έχουν ίσες αρμοδιότητες και να υπάρχει συνεργασία. Υπάρχει η δυνατότητα από μόνοι τους να χωρίσουν τις δραστηριότητες, ωστόσο αν ο εκπαιδευτικός το κρίνει σκόπιμο το πραγματοποιεί αυτός. Επίσης κάθε ομάδα θα πρέπει να έχει το δικό της Η/Υ καθώς και να υπάρχει η δυνατότητα σύνδεσης στο διαδίκτυο, για να μπορούν οι μαθητές να χρησιμοποιήσουν το Tinkercad και την πλατφόρμα του Arduino. Καλό θα ήταν να έχουν πρόσβαση και σε κάποιες σελίδες προγραμματισμού και στο βιβλίο πληροφορικής στο διαδίκτυο.

12.1 Αξιολόγηση

Για να αξιολογήσουμε τους μαθητές θα χρειαστούμε μία φόρμα. Τα στοιχεία θα τα πάρουμε κατά τη διάρκεια του μαθήματος για την συμπλήρωση της φόρμας. Ο εκπαιδευτικός θα ήταν καλό να συλλέγει σημειώσεις που αφορούν την επίδοση και την συμπεριφορά των μαθητών κατά την διάρκεια του μαθήματος. Κατά την ολοκλήρωση του μαθήματος θα πρέπει να καταγράψει αυτά τα στοιχεία στην φόρμα. Χρησιμοποιώντας αυτή την τεχνική μπορεί να δει που έχει κενά σημεία στην διδασκαλία του και μπορεί να τα βελτιώσει. Πιο κάτω σας παραθέτω το φύλλο αξιολόγησης των μαθητών :



Φύλλο Αξιολόγησης Μαθητών

1. Έμαθαν την έννοια των συναρτήσεων σε ένα πρόγραμμα;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
2. Δυσκολεύτηκαν στην υλοποίηση κάποιου προγράμματος;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
3. Κατανόησαν την διαφορά του void setup() από το void loop();	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
4. Τους φάνηκε διασκεδαστική η μέθοδος επίλυσης δραστηριοτήτων μέσω των κατασκευών στην πλατφόρμα Arduino;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
5. Έμαθαν την έννοια της ομαδικότητας ;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι

80

13.2 Το επιμορφωτικό σενάριο

1^η Διδακτική ώρα

Κατά την έναρξη του μαθήματος πραγματοποιείται μια σύντομη επανάληψη στις εντολές και τις συναρτήσεις του δομημένου προγραμματισμού.



2^η Διδακτική ώρα

Στην δεύτερη διδακτική ώρα οι μαθητές ανοίγουν την πλατφόρμα του Tinkercad και του Arduino και βλέπουν τις λειτουργίες τους. Εν συνεχεία έρχονται σε επαφή με την πλακέτα του Arduino και ανακαλύπτουν πως συνδέεται στον Η/Υ και τι περιφερειακά θα χρειαστούν για την εκπόνηση της άσκησης.

3^η Διδακτική ώρα

Στην 3^η Διδακτική ώρα οι μαθητές ξεκινάν να λύνουν το φύλλο εργασίας. Αρχικά λύνουν μία θεωρητική άσκηση για έξτρα επανάληψη. Και μετά ξεκινούν να λύνουν την επόμενη άσκηση όπου θα συζητήσουν για τεχνικές επίλυσης και θα κάνουν διάφορες υποθέσεις για την λύση της άσκησης.

4^η Διδακτική ώρα

Ο εκπαιδευτικός βοηθά στο να κυλήσει ομαλά η δραστηριότητα που έχει αναθέσει και οι μαθητές παράλληλα παρατηρούν τις λειτουργίες και την σύνταξη του δομημένου προγραμματισμού, τόσο στην πλατφόρμα Tinkercad καθώς και στο Arduino.

5^η και 6^η Διδακτική ώρα

Στις δύο τελευταίες διδακτικές ώρες οι μαθητές θα ολοκληρώσουν τον προγραμματισμό του κώδικα και θα περάσουν ό,τι έχουν κάνει στην πλατφόρμα του Tinkercad και εν συνεχεία στο Arduino. Έπειτα θ' ακολουθήσει συζήτηση στην τάξη με τους μαθητές για τυχόν απορίες και δυσκολίες. Επιπλέον θα γίνει σύγκριση των αποτελεσμάτων.

81

14.1 Προτάσεις για παραπάνω δραστηριότητες

Το συγκεκριμένο εκπαιδευτικό σενάριο μας δίνει την ευκαιρία να το συνεχίσουμε και αφού το έχουμε ολοκληρώσει. Έτσι λοιπόν οι μαθητές έχουν την δυνατότητα να βρουν ακόμη ένα τέτοιο παιχνίδι της ίδιας φιλοσοφίας ή να κάνουν πάνω στο υπάρχον μια επέκταση για παραπάνω δυσκολία. Εν συνεχεία θα αναζητήσουν τρόπους επίλυσης για να αποφύγουν τα λάθη κατά την διάρκεια της καινούργιας άσκησης. Τέλος θα πρέπει να κατασκευάσουν τον καινούργιο αλγόριθμο και να τον εφαρμόσουν πρώτα στο Tinkercad και στην συνέχεια στο Arduino.

15.1 Χρήση εξωτερικών πηγών

Χρήση του [Arduino Tutorial](https://www.arduino.cc/en/Tutorial/HomePage) (<https://www.arduino.cc/en/Tutorial/HomePage>)
https://en.wikipedia.org/wiki/Extreme_programming



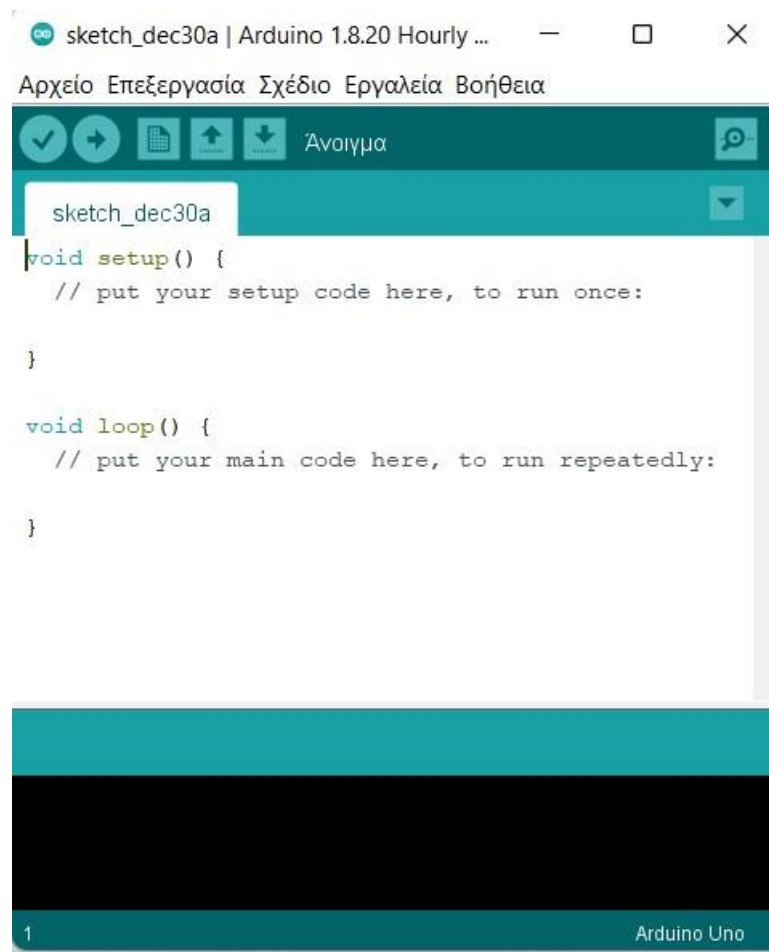
16.1 Φύλλα εργασίας

Φύλλα εργασίας μαθητών

**Θέμα: Παιχνίδι μνήμης με τη χρήση του
Tinkercad και του Arduino στη Γ' Λυκείου**

Όνοματεπώνυμο:.....

Τάξη:.....



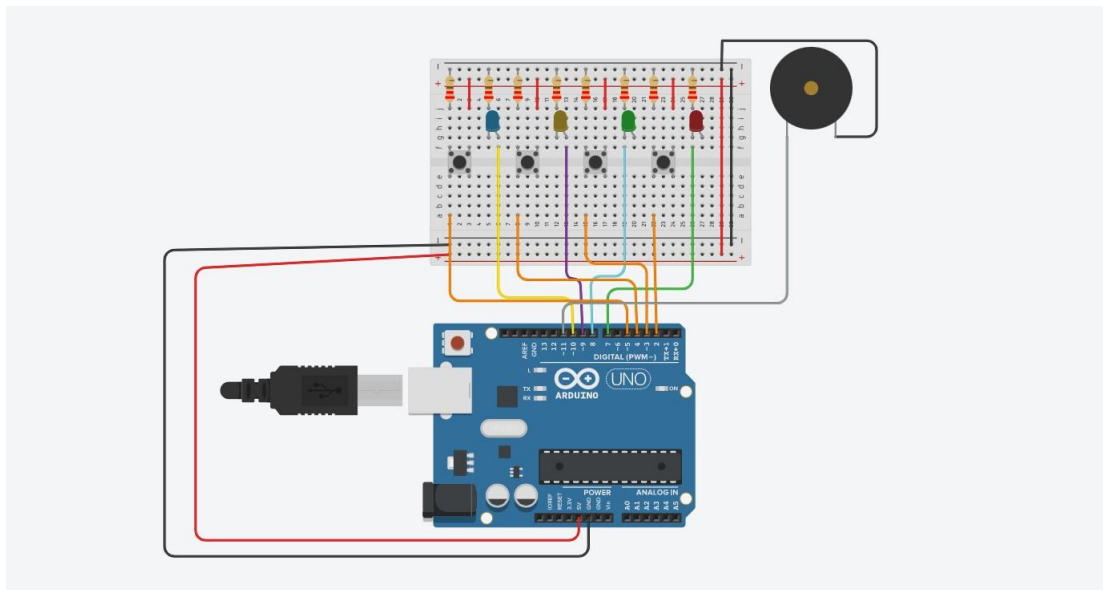
Εικόνα 25 - Περιβάλλον λογισμικού Arduino

Για να ανοίξουμε το λογισμικό του Arduino θα πρέπει να πάμε στην Έναρξη -> Όλα τα προγράμματα -> Arduino.



Για την υλοποίηση της άσκησης θα χρειαστούμε:

- ✚ 1 Πλακέτα Arduino
- ✚ 4 Λαμπτήρες Led
- ✚ 4 Κουμπιά
- ✚ 8 Αντιστάσεις των 220 ohm
- ✚ 1 Buzzer
- ✚ Καλώδια διασύνδεσης



Εικόνα 26 - Παιχνίδι μνήμης στην πλατφόρμα του Tinkercad

Κώδικας υλοποίησης διδακτικού σεναρίου

// Constants

```
const int button1 = 2; // Το πρώτο κουμπί ελέγχει το μπλέ Led
const int button2 = 3; // Το δεύτερο κουμπί ελέγχει το κίτρινο Led
const int button3 = 4; // Το τρίτο κουμπί ελέγχει το πράσινο Led
const int button4 = 5; // Το τέταρτο κουμπί ελέγχει το κόκκινο Led
const int led1 = 10; // Blue LED
const int led2 = 9; // Yellow LED
const int led3 = 8; // Green LED
const int led4 = 7; // Red LED
const int buzzer = 11; // Εξόδος του Buzzer
const int tones [] = {1915, 1700, 1519, 1432, 2700}; // Οι τόνοι όταν πατάμε τα Led
```

Διπλωματική Εργασία



– Το τελευταίο είναι όταν αποτυγχάνουμε.

// Μεταβλητές

```
int buttonState[] = {0,0,0,0}; // Η τρέχουσα κατάσταση του κουμπιού
int lastButtonState[] = {0,0,0,0}; // Η προηγούμενη κατάσταση του κουμπιού
int buttonPushCounter[] = {0,0,0,0};
void playTone (int tone, int duration) {
for (long i = 0; i < duration * 1000L; i += tone * 2) {
digitalWrite(buzzer, HIGH); delayMicroseconds(tone);
digitalWrite(buzzer, LOW); delayMicroseconds(tone);
}
}
void setup() {
// Αρχικοποίηση των εισόδων:
randomSeed (analogRead (0));
pinMode(button1, INPUT);
pinMode(button2, INPUT);
pinMode(button3, INPUT);
pinMode(button4, INPUT);
// Αρχικοποίηση των εξόδων:
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
pinMode(buzzer, OUTPUT);
//Αρχικοποίηση της σειριακής επικοινωνίας για το debugging:
Serial.begin(9600);
}
int game_on = 0;
int wait = 0;
int currentlevel = 1; /* Εδώ κρατάμε το επίπεδο (επίσης ο αριθμός των
πατημάτων των κουμπιών για να πάμε στο επόμενο επίπεδο) */
long rand_num = 0; /*Κάνει αρχικοποίηση μιας τυχαίας μεταβλητής από το 0-
100. */
```



```
int rando = 0; /*Αρχικοποιεί μια τυχαία ακέραια όταν το loopgame_on.Μετά  
πάει από 1-4. */  
int butwait = 500; //Ο χρόνος αναμονής για την επόμενη είσοδο κουμπιού.  
int ledtime = 500; /*το χρονικό διάστημα που κάθε LED αναβοσβήνει όταν πατηθεί το  
κουμπί. */  
int n_levels = 10; //Ο αριθμός των επιπέδων μέχρι να κερδηθεί το παιχνίδι  
int pinandtone = 0; /*Αυτή η ακέραια χρησιμοποιείται όταν εμφανιστεί η  
ακολουθία */  
int right = 0; /*Αυτή η μεταβλητή θα πρέπει να είναι ένα για να πάμε στο  
επόμενο επίπεδο. */  
int speedfactor = 5; /*Αυτή είναι η τελική ταχύτητα των Led και των ήχων για  
το τελευταίο επίπεδο. Αυξάνει όταν κερδίζονται περισσότερα παιχνίδια. */  
int leddelay = 200; /*Αρχικοποίηση του χρόνου για το Led. Αυτός μειώνεται όσο  
τα επίπεδα ανεβαίνουν. */  
  
void loop() {  
int n_array[n_levels];  
int u_array[n_levels];  
int i;  
//Καθαρίζει από κοινού "n_array" και "u_array" και ξεκινάει καινούργιο  
παιχνίδι.  
if (game_on == 0){  
for(i=0; i<n_levels; i=i+1){  
n_array[i]=0;  
u_array[i]=0;  
rand_num = random(1,200);  
if (rand_num <= 50)  
rando=0;  
else if (rand_num>50 && rand_num<=100)  
rando=1;  
else if (rand_num>100 && rand_num<=150)  
rando=2;  
else if (rand_num<=200)
```



```
rando=3;
//Σώζει έναν τυχαίο αριθμό στην n_array
  n_array[i]=rando;
}
game_on = 1;
}
//Δείχνει στον χρήστη την τρέχουσα σειρά.
  if (wait == 0){
delay (200);
i = 0;
for (i = 0; i < currentlevel; i= i + 1){
leddelay = ledtime/(1+(speedfactor/n_levels)*(currentlevel - 1));
  pinandtone = n_array[i];
digitalWrite(pinandtone+7, HIGH);
playTone(tones[pinandtone], leddelay);
digitalWrite(pinandtone+7, LOW);
delay(100/speedfactor);
}
wait = 1;
}
i = 0;
int buttonchange = 0;
int j = 0; // Εδώ είναι η τρέχουσα θέση της σειράς.
  while (j < currentlevel){
while (buttonchange == 0){ for (i = 0; i < 4; i = i + 1){
buttonState[i] = digitalRead(i+2);
  buttonchange = buttonchange + buttonState[i];
}
}
for (i = 0; i < 4; i = i + 1){
if (buttonState[i] == HIGH) {
  digitalWrite(i+7, HIGH);
  playTone(tones[i], ledtime);
```




```
digitalWrite(i+7, LOW);
wait = 0;
u_array[j]=i;
  buttonState[i] = LOW;
  buttonchange = 0;
}
}
if (u_array[j] == n_array[j]){
  j++;
  right = 1;
}
else{
  right = 0;
  i = 4;
  j = currentlevel;
  wait = 0;
}
}
if (right == 0){
  delay(300);
  i = 0;
  game_on = 0;
  currentlevel = 1;
  for (i = 0; i < 4; i = i + 1){
    digitalWrite(i+7, HIGH);
  }
  playTone(tones[4], ledtime);
  for (i = 0; i < 4; i = i + 1){
    digitalWrite(i+7, LOW);
  }
  delay (200);
  for (i = 0; i < 4; i = i + 1){
    digitalWrite(i+7, HIGH);
```



```
}  
playTone(tones[4], ledtime);  
for (i = 0; i < 4; i = i + 1){  
    digitalWrite(i+7, LOW);  
}  
delay(500);  
game_on = 0;  
}  
//Αν βάλουμε την σωστή ακολουθία ανεβαίνει επίπεδο.  
if (right == 1){  
currentlevel++;  
    wait = 0;  
}  
//Αν ολοκληρωθεί το παιχνίδι  
if (currentlevel == n_levels){  
    delay(500);  
// Εδώ κρατάμε τον ήχο της επιτυχίας:  
int notes[] = {2, 2, 2, 2, 0, 1, 2, 1, 2};  
int note = 0;  
int tempo[] = {200, 200, 200, 400, 400, 400, 200, 200, 600};  
int breaks[] = {100, 100, 100, 200, 200, 200, 300, 100, 200};  
for (i = 0; i < 9; i = i + 1){  
    note = notes[i];  
    digitalWrite(note+7, HIGH);  
playTone(tones[note], tempo[i]);  
    digitalWrite(note+7, LOW);  
    delay(breaks[i]); } } }
```



3.4.2 Το δεύτερο διδακτικό σενάριο

Μέτρηση της θερμοκρασίας με την βοήθεια αισθητήρα LM και απεικόνιση των τιμών σε RGB Led.

1.2 Τίτλος εκπαιδευτικού σεναρίου

Μέτρηση της θερμοκρασίας με την βοήθεια αισθητήρα LM και απεικόνιση των τιμών σε RGB Led.

2.2 Εκτιμώμενη διάρκεια εκπαιδευτικού σεναρίου

Για να πραγματοποιηθεί το συγκεκριμένο σενάριο θα χρειαστούμε 5 διδακτικές ώρες. Θα χωριστούν από τον εκπαιδευτικό της τάξης, σε 2 ώρες θεωρία και 3 ώρες σε υλοποίηση των ασκήσεων.

3.2 Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών – Απαιτούμενες γνώσεις.

Το εν λόγω διδακτικό σενάριο απευθύνετε σε μαθητές Γ Λυκείου, όπου είναι απαραίτητο να έχουν Προαπαιτούμενες γνώσεις έτσι ώστε να το εκπονήσουν. Επιπλέον το συγκεκριμένο διδακτικό σενάριο ακολουθεί το αναλυτικό πρόγραμμα σπουδών της Γ Λυκείου και το μάθημα της πληροφορικής. Εντάσσεται προς το τέλος το κεφαλαίου του δομημένου προγραμματισμού και οι μαθητές θα πρέπει να είναι σε θέση να γνωρίζουν βασικές εντολές, να μπορούν να κάνουν καλή χρήση του Η/Υ και επιπλέον να γνωρίζουν την πλατφόρμα του Arduino και του Tinkercad.

4.2 Σκοποί και στόχοι του εκπαιδευτικού σεναρίου

Σκοπός του εκπαιδευτικού σεναρίου είναι οι μαθητές να μάθουν την λειτουργία του αισθητήρα θερμοκρασίας LM. Αφού καταλάβουν το πως λειτουργεί ο αισθητήρας να υλοποιήσουν ένα κύκλωμα με RGB Led, για να μπορούν να παρομοιάζουν τις τιμές του αισθητήρα και με την βοήθεια του εκπαιδευτικού και των ενδεδειγμένων ασκήσεων να συντάξουν τον κώδικα λειτουργίας.

Στόχοι εκπαιδευτικού σεναρίου:

Με την ολοκλήρωση του εκπαιδευτικού σεναρίου οι μαθητές θα πρέπει:



- ✚ Να δοκιμάσουν διαφορετικές τεχνικές επίλυσης στο πρόβλημα.
- ✚ Να μάθουν να δημιουργούν έξυπνες κατασκευές χρησιμοποιώντας την πλατφόρμα του Arduino και του Tinkercad.
- ✚ Να ανακαλύψουν την έννοια της ομαδοσυνεργατικότητας.
- ✚ Να προσαρμόσουν τους αλγόριθμους σε διαφορετικά προγραμματιστικά περιβάλλοντα (Arduino, Tinkercad).

5.2 Περιγραφή εκπαιδευτικού σεναρίου

Οι μαθητές χρησιμοποιώντας διάφορες τεχνικές θα δημιουργήσουν ένα εκπαιδευτικό σενάριο το οποίο θα περιλαμβάνει έναν αισθητήρα θερμοκρασίας τύπου LM, ένα RGB Led και τις απαραίτητες συνδεσμολογίες με τον μικροελεγκτή Arduino. Η δημιουργία του εκπαιδευτικού σεναρίου θα βοηθήσει τους μαθητές να κατανοήσουν την σύνταξη αλλά και την λειτουργία του δομημένου προγραμματισμού. Επιπλέον θα έχουν την ευκαιρία να δημιουργήσουν απλές κατασκευές στην πλατφόρμα του Tinkercad και του Arduino. Οι μαθητές θα εργαστούν σε ομάδες έπειτα από υποδείξεις του επιβλέποντα εκπαιδευτικού, κατά συνέπεια θα αναπτύξουν το αίσθημα της ευθύνης, της υπευθυνότητας αλλά και την ομαδοσυνεργατικότητα. Ο τρόπος αυτός θα τους οδηγήσει στην ευκολότερη επίλυση του προβλήματος.

Υλοποιώντας το σενάριο αυτό οι μαθητές θα έχουν την δυνατότητα να κάνουν χρήση διαφόρων διδακτικών προσεγγίσεων, με αποτέλεσμα η κατανόηση του μαθήματος να γίνει ευκολότερη. Επιπλέον μαθαίνουν να προσαρμόζονται πιο εύκολα στους νέους τρόπους εκπαίδευσης. Οι προσεγγίσεις που θα ακολουθήσουμε είναι οι εξής:

- ✚ **Θεωρητικό μέρος:** Σ' αυτό το μέρος οι εκπαιδευόμενοι θα πραγματοποιήσουν μία επανάληψη στον δομημένο προγραμματισμό, στην λειτουργία του αλλά και στην σύνταξή του. Επιπρόσθετα θα έχουν μια πρώτη επαφή με την πλακέτα Arduino αλλά και μία υπενθύμιση στην πλατφόρμα του Tinkercad.
- ✚ **Πρακτικό μέρος:** Σ' αυτό το μέρος οι εκπαιδευόμενοι θα ξεκινήσουν με την επίλυση του φύλλου εργασίας, κάνοντας χρήση αυτών που έχουν διδαχθεί στο θεωρητικό μέρος. Θα γίνει εκμάθηση του λογισμικού του Arduino λύνοντας διάφορους αλγόριθμους, ενώ παράλληλα μ' αυτό θα βελτιώσουν και τις γνώσεις τους στον δομημένο προγραμματισμό.



- ✚ **Ομαδικό μέρος:** Σ' αυτό το μέρος οι εκπαιδευόμενοι θα πρέπει να εργαστούν σε ομάδες, για να ολοκληρώσουν το εκπαιδευτικό σενάριο. Θα αναπτύξουν το αίσθημα της ομαδοσυνεργατικότητας και της συνέπειας. Ο κάθε εκπαιδευόμενος θα αναλάβει το δικό του κομμάτι στο εκπαιδευτικό σενάριο, καλλιεργώντας έτσι το αίσθημα της υπευθυνότητας. Επειδή οι εκπαιδευόμενοι εργάζονται σε ομάδες αποκτούν μεγαλύτερο ενδιαφέρον για το μάθημα, ανταλλάζουν απόψεις και αποκτούν περισσότερες γνώσεις για το αντικείμενο.
- ✚ **Εξερευνητικό μέρος:** Σ' αυτό το μέρος οι εκπαιδευόμενοι αναζητούν πιθανές λύσεις για το εκπαιδευτικό σενάριο, στην δραστηριότητα την οποία εκπονούν. Επιπλέον θα πραγματοποιήσουν δοκιμές σε διάφορες τεχνικές επίλυσης μέχρι να καταλήξουν στην λύση του εκπαιδευτικού σεναρίου.

6.2 Επιστημολογική προσέγγιση και εννοιολογική ανάλυση – θέματα θεωρίας του εκπαιδευτικού

Το παραπάνω εκπαιδευτικό σενάριο εφαρμόζει την τακτική του Extreme Programming, όπου χρησιμοποιήθηκε από τον Kent Beck. Μία μέθοδος η οποία είναι βασισμένη στον προγραμματισμό ανά ζεύγη, βοηθώντας να έχουμε καλύτερη εικόνα για τον κώδικα μας. Χρησιμοποιώντας αυτή τη μέθοδο η παραγωγικότητα των εκπαιδευόμενων αυξάνεται και αποκτούν την ευχέρεια να εκπονήσουν δυσκολότερα σενάρια. Η μέθοδος του Extreme Programming βασίζεται σε τέσσερις βασικές λειτουργίες: Τον έλεγχο, τον προγραμματισμό, τον σχεδιασμό της εμφάνισης του προγράμματος και την κατανόηση των αναγκών του χρήστη.

7.2 Χρήση Η/Υ και ψηφιακών μέσων για το εκπαιδευτικό σενάριο

Για την εκπόνηση του συγκεκριμένου εκπαιδευτικού σεναρίου είναι αναγκαίο οι εκπαιδευόμενοι, να έχουν πραγματοποιήσει εγκατάσταση του λογισμικού του Arduino στον Η/Υ. Επιπλέον θα πρέπει να έχουν πρόσβαση στο διαδίκτυο καθώς πριν την υλοποίηση στην πλατφόρμα θα γίνει προσομοίωση στο Tinkercad. Η κάθε ομάδα είναι αναγκαίο να έχει τους δικούς της αισθητήρες αλλά και πλακέτα.

8.2 Πρόβλεψη δυσκολιών μαθητών στο εκπαιδευτικό σενάριο

Το παραπάνω εκπαιδευτικό σενάριο κάνει χρήση νέων μεθόδων διδασκαλίας, κατά συνέπεια οι εκπαιδευόμενοι μπορεί να συναντήσουν δυσκολίες στην εκμάθηση αλλά και στην πρακτική εφαρμογή του. Χρησιμοποιώντας τα παραδείγματα βελτιώνουν



την ικανότητα τους στην υλοποίηση των κυκλωμάτων καθώς και στον προγραμματισμό. Οι πλατφόρμες του Tinkercad και του Arduino παρόλο που είναι στην αγγλική γλώσσα είναι εύκολες στην χρήση ακολουθώντας τα βήματα του οδηγού, αλλά και την βοήθεια του εκπαιδευτή. Τα βήματα είναι προσαρμοσμένα σε μαθητές της Γ Λυκείου. Επιπλέον η εγκατάσταση του Arduino είναι απλή και μπορεί να πραγματοποιηθεί απ' όλους τους εκπαιδευόμενους.

9.2 Διδακτικό σύμβολαιο

Ο εκπαιδευτικός κατά την διάρκεια του εκπαιδευτικού σεναρίου θα πρέπει να είναι υποστηρικτικός προς τους εκπαιδευόμενους. Θα τους παρέχει κατευθυντήριες γραμμές έτσι ώστε οι εκπαιδευόμενοι να οδηγούνται μόνοι τους στην λύση. Αυτή η αλληλεπίδραση θα έχει ως αποτέλεσμα να δημιουργηθεί ομαδικό πνεύμα στην τάξη, και οι εκπαιδευόμενοι θα οδηγηθούν πιο γρήγορα στην επίλυση του εκπαιδευτικού σεναρίου.

10.2 Υποκείμενη θεωρία μάθησης

Οι εκπαιδευόμενοι κάνοντας χρήση στοιχείων του δομημένου προγραμματισμού, θέτουν ως στόχο την επίλυση του εκπαιδευτικού σεναρίου ενώ παράλληλα ο εκπαιδευτικός του χωρίζει σε ομάδες. Τα φύλλα εργασίας που θα τους ανατεθούν είναι κλιμακωτής δυσκολίας και τα βήματα οδηγούν τον εκπαιδευόμενο στην λύση. Ο χωρισμός των ομάδων γίνεται με βάση το γνωστικό επίπεδο του κάθε εκπαιδευόμενου από τον εκπαιδευτικό. Με τον τρόπο αυτό έχουν την ευκαιρία για ανταλλαγή ιδεών και τεχνικών προγραμματισμού. Η διαδικασία αυτή οδηγεί το μάθημα σε ένα πολύ ευχάριστο κλίμα όπου όλοι οι εκπαιδευόμενοι απολαμβάνουν.

11.2 Οργάνωση της τάξης

Για την υλοποίηση του συγκεκριμένου εκπαιδευτικού σεναρίου είναι αναγκαίο οι εκπαιδευόμενοι να είναι χωρισμένοι σε ομάδες. Ο διαχωρισμός πραγματοποιείται από το εκπαιδευτικό προκειμένου να δοθεί η απαραίτητη ισορροπία, που αφορά το γνωστικό επίπεδο κάθε εκπαιδευόμενου. Η συνεργασία και οι ίσες αρμοδιότητες είναι τα κύρια χαρακτηριστικά τα οποία πρέπει να διαθέτει κάθε ομάδα. Επιπλέον ο εκπαιδευτικός έχει την δυνατότητα να αφήσει του εκπαιδευόμενους να χωρίσουν μόνοι τους τις ομάδες, πραγματοποιώντας οποιαδήποτε αλλαγή αν το κρίνει σκόπιμο. Η κάθε ομάδα διαθέτει τον δικό της Η/Υ με σύνδεση στο διαδίκτυο, έτσι ώστε οι



εκπαιδευόμενοι να έχουν την δυνατότητα να χρησιμοποιήσουν την πλατφόρμα του Tinkercad. Έπειτα αν ο εκπαιδευτικός κρίνει αναγκαίο όλο το εκπαιδευτικό σενάριο περνάει στην πλατφόρμα του Arduino. Οι εκπαιδευόμενοι έχουν συνεχώς την υποστήριξη του εκπαιδευτικού και μπορούν οποιαδήποτε πληροφορία χρειαστούν να τη λάβουν από το βιβλίο πληροφορικής του σχολείου ή το διαδίκτυο.

12.2 Αξιολόγηση

Οι εκπαιδευόμενοι κατά την διάρκεια του εκπαιδευτικού σεναρίου θα περάσουν ένα είδος αξιολόγησης. Τα στοιχεία που είναι αναγκαία για τον εκπαιδευτικό θα παρθούν από την συμπλήρωση μιας φόρμας. Επιπλέον κατά την διάρκεια του σεναρίου ο εκπαιδευτικός θα συλλέγει σημειώσεις που είναι αναγκαίες για την επίδοση κάθε μαθητή, καθώς και για την συμπεριφορά. Με την ολοκλήρωση του εκπαιδευτικού σεναρίου η κάθε φόρμα θα πρέπει να είναι συμπληρωμένη. Η συγκεκριμένη τεχνική θα βοηθήσει τον εκπαιδευτικό να ανακαλύψει πιθανά κενά στην διδασκαλία του και να τα βελτιώσει. Παρακάτω παρατεθείτε η φόρμα αξιολόγησης των εκπαιδευόμενων:

Φύλλο Αξιολόγησης Μαθητών

95

1. Έμαθαν την έννοια των συναρτήσεων σ' ένα πρόγραμμα;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
2. Δυσκολεύτηκαν στην υλοποίηση κάποιου προγράμματος;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
3. Κατανόησαν την διαφορά του void setup() από το void loop();	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι



4. Τους φάνηκε διασκεδαστική η μέθοδος επίλυσης δραστηριοτήτων μέσω των κατασκευών στην πλατφόρμα Arduino;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι
5. Έμαθαν την έννοια της ομαδικότητας ;	Ναι	Σχεδόν Ναι	Σχεδόν Όχι	Όχι

13.2 Το επιμορφωτικό σενάριο

1^η Διδακτική ώρα

Ξεκινώντας το εκπαιδευτικό σενάριο ο εκπαιδευτικός υλοποιεί μια σύντομη επανάληψη σε διάφορες εντολές και συναρτήσεις του δομημένου προγραμματισμού.

2^η Διδακτική ώρα

Οι εκπαιδευόμενοι κατά την δεύτερη διδακτική ώρα έρχονται σε μία πρώτη επαφή με την πλατφόρμα του Tinkercad και του Arduino. Ανακαλύπτουν διάφορες λειτουργίες των πλατφορμών αλλά και της πλακέτας Arduino. Έπειτα ο εκπαιδευτικός δείχνει τις απαραίτητες συνδεσμολογίες στους εκπαιδευόμενους.

3^η Διδακτική ώρα

Οι εκπαιδευόμενοι επιλύουν το φύλλο εργασίας που τους παρέχεται από τον εκπαιδευτικό. Η μέθοδος αυτή βοηθά και τους δύο για διαφορετικούς λόγους. Ο εκπαιδευτικός βγάζει χρήσιμα συμπεράσματα για τον πρόοδο των εκπαιδευόμενων. Οι εκπαιδευόμενοι αρχικά λύνουν μία θεωρητική άσκηση πραγματοποιώντας μια μικρή επανάληψη. Εν συνεχεία επιλύουν την επόμενη άσκηση του φύλλου εργασίας, όπου αυτό το μέρος περιέχει πιο πρακτικό κομμάτι πραγματοποιώντας μία συζήτηση με την ομάδα τους για τον τρόπο επίλυση της άσκησης.



4^η Διδακτική ώρα

Ο εκπαιδευτικός κατά την τέταρτη διδακτική ώρα καθοδηγεί τις ομάδες να εκπονήσουν το εκπαιδευτικό σενάριο. Επιπλέον ανακαλύπτουν τις λειτουργίες και την σύνταξη του δομημένου προγραμματισμού στο Tinkercad και στο Arduino.

5^η και 6^η Διδακτική ώρα

Κατά την διάρκεια των δύο τελευταίων διδακτικών ωρών οι εκπαιδευόμενοι διευθετούν τα τελευταία θέματα που αφορούν τον προγραμματισμό. Εν συνεχεία πραγματοποιούν έλεγχο ό,τι έχουν κάνει στο Tinkercad, και αν ο εκπαιδευτικός το κρίνει σκόπιμο γίνεται η μεταφορά του σεναρίου στην πλατφόρμα του Arduino.

14.2 Προτάσεις για παραπάνω δραστηριότητες

Με το παραπάνω εκπαιδευτικό σενάριο έχουμε την ευκαιρία να το συνεχίσουμε και μετά την ολοκλήρωση του και να προσθέσουμε και άλλες δραστηριότητες. Οι εκπαιδευόμενοι έχουν την δυνατότητα να προσθέσουν και άλλους αισθητήρες, πλην του αισθητήρα θερμοκρασίας και να ανεβάσουν το επίπεδο δυσκολίας τόσο στην συνδεσμολογία αλλά και στον κώδικα. Επιπλέον οι εκπαιδευόμενοι βρίσκουν διάφορους τρόπους για να επιλύσουν την νέα δραστηριότητα, με σκοπό την αποφυγή λαθών. Τέλος η σχεδίαση του νέου αλγορίθμου είναι αναγκαία και προσαρμόζεται πρώτα στο Tinkercad και αν ο εκπαιδευτικός το κρίνει αναγκαίο έπειτα στην πλατφόρμα του Arduino.

15.2 Χρήση εξωτερικών πηγών

Χρήση του [Arduino Tutorial](https://www.arduino.cc/en/Tutorial/HomePage) (https://www.arduino.cc/en/Tutorial/HomePage)

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

16.2 Φύλλα εργασίας

Φύλλα εργασίας μαθητών

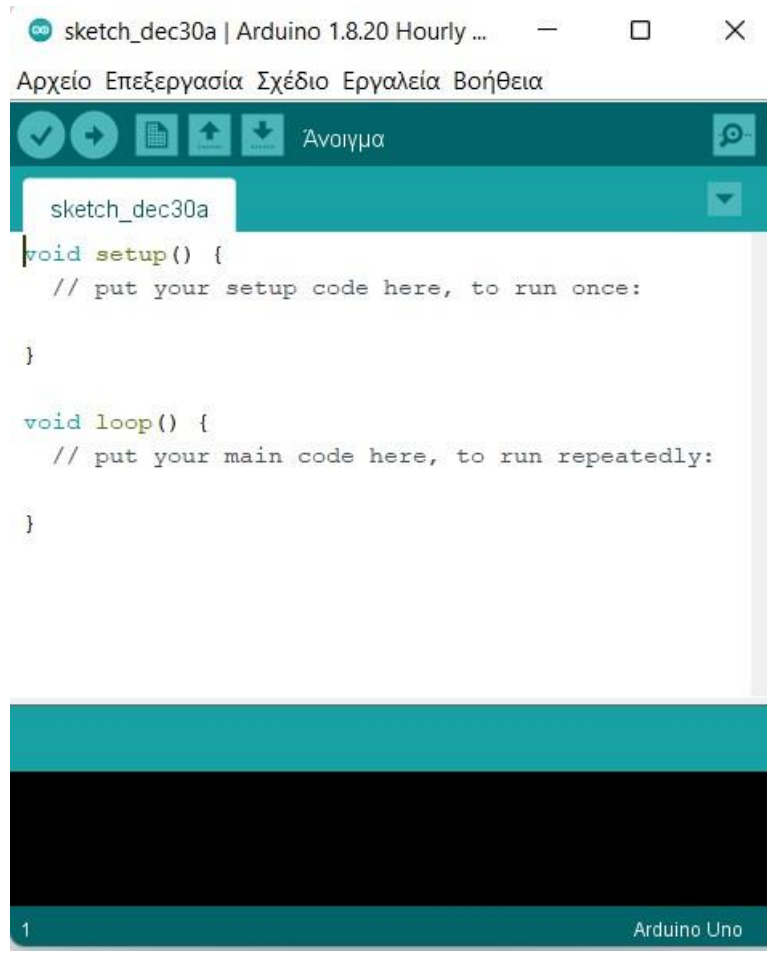
Θέμα: Μέτρηση της θερμοκρασίας με την βοήθεια αισθητήρα LM και απεικόνιση των τιμών σε RGB Led.

Tinkercad και του Arduino στη Γ' Λυκείου

Ονοματεπώνυμο:.....



Τάξη:.....



Εικόνα 27 - Περιβάλλον λογισμικού Arduino2

Για να ανοίξουμε το λογισμικό του Arduino θα πρέπει να πάμε στην Έναρξη -> Όλα τα προγράμματα -> Arduino.

Βασικές λειτουργίες του προγράμματος



: Επιβεβαίωση του κώδικα



: Φόρτωμα κώδικα στην πλατφόρμα Arduino



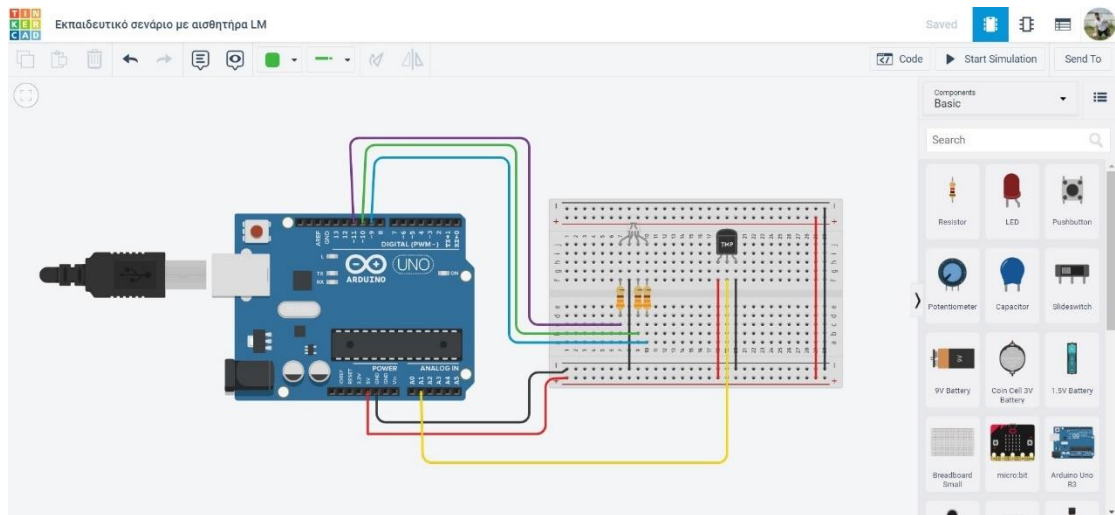
: Νέο έγγραφο



: Άνοιγμα αρχείου



: Αποθήκευση αρχείου



Εικόνα 28 - Παράδειγμα αισθητήρα θερμοκρασίας LM

Κώδικας υλοποίησης διδακτικού σεναρίου

```
int redPin = 11; // Σύνδεση του Red led στο pin11
int greenPin = 10; // Σύνδεση του Green led στο pin12
int bluePin = 9; // Σύνδεση του Blue led στο pin
float temp_c; // Αποθήκευση της τιμής της Θερμοκρασίας
int sensor_val; // Αποθήκευση της τιμής της Τάσης από τον αισθητήρα (0-1023)
int temp_pin = 1; // Κανάλι εισόδου του ADC.
int lcd;
void setup()
{
  Serial.begin(9600);
  pinMode(redPin, OUTPUT); // Ορίζονται τα pins ως έξοδοι
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  analogReference(INTERNAL); /* Ορισμός της τάσης αναφοράς του ADC στα
1,1
Volts με χρήση εσωτερικής στάθμης αναφοράς του μικροελεγκτή ( μείωση από
τα 5 στα 1,1
Volts για καλύτερη διακριτική ακρίβεια – 4,9 mV σε 1 mV) */
}
void loop()
{
```



```
sensor_val = analogRead(temp_pin); /* Διάβασμα της τάσης που δίνει το LM35
από
τον ADC (κανάλι εισόδου 1 – pin A1) και απόδοση της τιμής (0 – 1023) στην
μεταβλητή
sensor_val. *

temp_c = (sensor_val / 9.31); // Μετατροπή σε βαθμούς Κελσίου.
// Επιλογή χρώματος ανάλογα με την τιμή της temp_c
if (temp_c >= 40.00)
{ setColor(225, 0, 0); // Κόκκινο χρώμα
}
else if (temp_c >= 32.00 && temp_c < 40.00)
{ setColor(225, 35, 0); // Πορτοκαλί χρώμα
}
else if (temp_c >= 25.00 && temp_c < 32.00)
{ setColor(225, 105, 0); // Κίτρινο χρώμα
}
else if (temp_c >= 18.00 && temp_c < 25.00)
{ setColor(0, 225, 225); // Κυανό χρώμα
}
else if (temp_c >= 11.00 && temp_c < 18.00)
{ setColor(0, 0, 225); // Μπλέ χρώμα
}
else if (temp_c >= 4.00 && temp_c < 11.00)
{ setColor(225, 0, 100); // Μοβ Χρώμα
}
else
{ setColor(225, 225, 225); // Άσπρο χρώμα
}
delay(1000);
}
void setColor(int red, int green, int blue)
{
```



```
#ifndef COMMON_ANODE
red = 255 - red;
green = 255 - green;
blue = 255 - blue;
#endif
analogWrite(redPin, red);
analogWrite(greenPin, green);
analogWrite(bluePin, blue);
Serial.println(temp_c);
}
```

Υποστηρικτικό θεωρητικό υλικό για τον αισθητήρα θερμοκρασίας

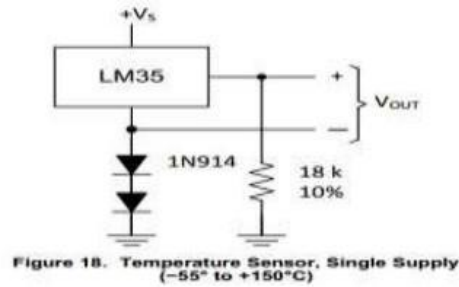
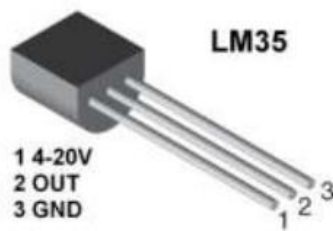
Στο συγκεκριμένο εκπαιδευτικό σενάριο θα χρησιμοποιήσουμε τον αισθητήρα θερμοκρασίας LM. Οι αισθητήρες αυτοί παρέχουν μεγάλη ακρίβεια. Η λειτουργία τους παρέχει την δυνατότητα να χρησιμοποιήσουν την τάση εξόδου γραμμικά ως προς την θερμοκρασία σε κλίμακα των βαθμών κελσίου. Το μεγάλο πλεονέκτημα του αισθητήρα έναντι σε άλλους αισθητήρες που είναι βαθμονομημένοι σε βαθμούς κέλβιν, είναι πως ο χρήστης δεν είναι αναγκαίο να πραγματοποιήσει μεγάλη αφαίρεση από την σταθερή τάση εξόδου, με σκοπό να ληφθεί η κλιμάκωση στον βαθμούς κελσίου.

Κατά συνέπεια από τον αισθητήρα LM μας δίνετε 10 mV ανά ένα βαθμό κελσίου. Έτσι για να μπορέσουμε να μετρήσουμε σε βαθμούς κελσίου θα πρέπει να χρησιμοποιήσουμε την τάση αναφοράς για τον ADC στα 1.1V. Αν η διακριτική ικανότητα του ADC είναι στα 10 bits τότε έχουμε:

$$\color{red}{\oplus} 2^{10} = 1024 \text{ στάθμες με } (1.1 / 1024) = 1.0742 \text{ m V διακριτική ικανότητα του ADC.}$$

Χρησιμοποιώντας το LM έχουμε 10 m V ανά ένα βαθμό της κλίμακας κελσίου. Οπότε κάθε μετατροπή της στάθμης του ADC (1.0742 m V) αναλογεί σε $(1.0742 \text{ m V} / 10 \text{ m V}) = 0.10742 = (1/ 9.31)$ βαθμούς κελσίου.

Πολύ σημαντικό επιπλέον γι' αυτό τον αισθητήρα είναι πως δεν έχει την ανάγκη να πραγματοποιηθεί καλιμπράρισμα ή γενικότερα άλλες ρυθμίσεις για να έχει την απαιτούμενη ακρίβεια. Το εύρος των βαθμών κελσίου του αισθητήρα είναι από -55 έως 150 ανάλογα βέβαια την έκδοση.



Typical Applications

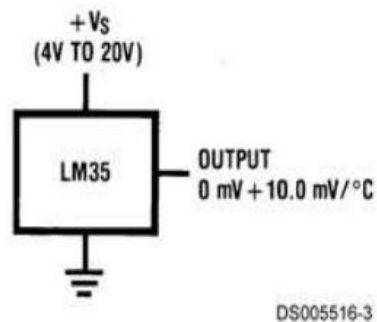


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

4.Ο μικροελεγκτής Micro:bit

4.1 Ιστορικά στοιχεία του μικροελεγκτή Micro:bit

Την χρονιά του 2013 στο Ηνωμένο Βασίλειο παρουσιάστηκε ένα νέο πρόγραμμα σπουδών. Το συγκεκριμένο πρόγραμμα τροποποιήθηκε από την κλασική διδασκαλία που έχουμε στην πληροφορική (Εξάσκηση σε δυνατότητες και σε εργαλεία του Η/Υ και παρουσίαση), σε εκμάθηση του προγραμματισμού μέσω της χρήσης. Κατά συνέπεια οι εκπαιδευόμενοι αντί να διδάσκονται πως να εργάζονται στον Η/Υ, θα διδάσκονται το πως να τον προγραμματίζουν σύμφωνα με τα θέλω τους.



Η δημιουργικότητα και η πρωτότυπη σκέψη θα αποκτήσουν πλέον μεγάλη βαρύτητα δίνοντας την ευκαιρία στους εκπαιδευόμενους να δουλεύουν όπως αυτοί θέλουν, ανακαλύπτοντας μόνοι τους τα προβλήματά τους και να δουλεύουν πάνω σ' αυτά μέχρις ότου τα λύσουν. Η διδασκαλία του προγραμματισμού θα διδάσκετε πλέον χωρίς να θεωρείται αυτοσκοπός, αλλά ως ένα εργαλείο που μας παρέχει ευρηματικές λύσεις. Έτσι χρησιμοποιώντας αυτή την προσέγγιση ο προγραμματισμός προσελκύει όλο και περισσότερα παιδιά.

Ο μικροελεγκτής του Micro:bit προέκυψε έπειτα από πρωτοβουλία του BBC, να δημιουργήσει μια πλακέτα όπου το κόστος της θα είναι χαμηλό και θα μπορεί να χρησιμοποιηθεί στην σχολική κοινότητα. Η χρήση του μας παρέχει μεγάλο εύρος δραστηριοτήτων από τον έλεγχο ρομπότ μέχρι και τον έλεγχο μουσικών οργάνων. Ωστόσο η ηλικιακή ομάδα που χρησιμοποιείται είναι μικρότερη από αυτή του Arduino. Ο κύριος λόγος είναι ότι οι συνδεσμολογίες του είναι πιο απλές και το μέγεθος της πλακέτας είναι μικρότερο. Στις ημέρες μας το Micro:bit έχει εισχωρήσει σε σχολεία όλου του κόσμου (Φινλανδία, Ισλανδία, Σρι Λάνκα, Σιγκαπούρη κ.α) και με γρήγορα βήματα εισέρχεται και στην Ελλάδα.

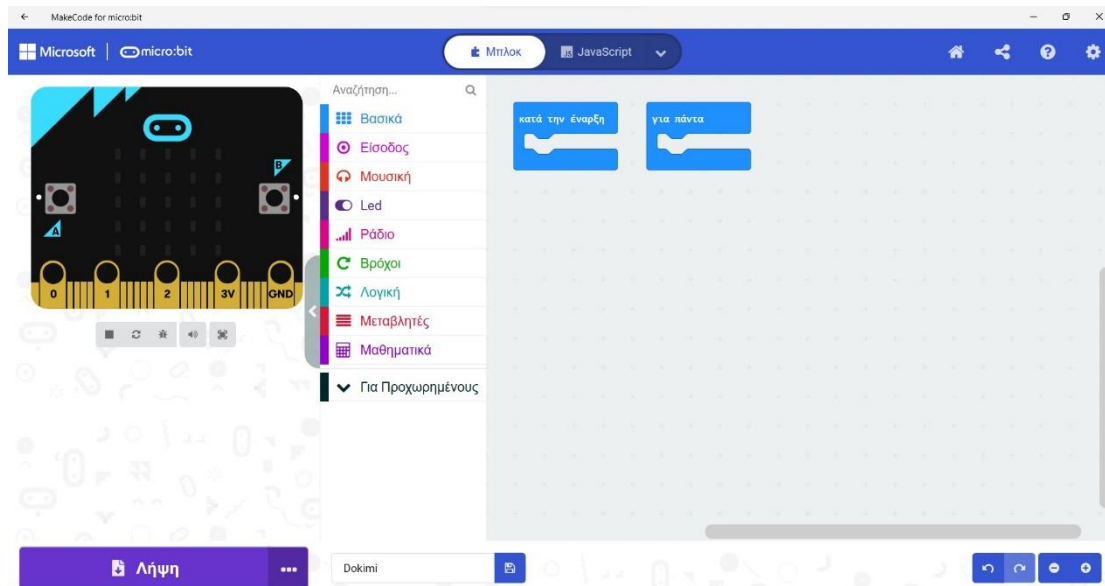
4.2 Γενικές πληροφορίες και οδηγίες χρήσης για το micro:bit

Το micro:bit δημιουργήθηκε από κάποιες εταιρίες με την βοήθεια του BBC της Μεγάλης Βρετανίας. Θεωρείται ένας μικροϋπολογιστής όπου κάνει χρήση βιβλιοθηκών ανοιχτού κώδικα, έχοντας ως συνέπεια την διευκόλυνση των μαθητών να διδαχθούν προγραμματισμό και να δημιουργήσουν διαδραστικά έργα τα οποία είναι περισσότερο οικεία στα παιδιά.

Το μέγεθος του μικροελεγκτή δεν ξεπερνάει σε διαστάσεις το μέγεθος μίας πιστωτικής κάρτας, χρησιμοποιώντας τον ARM M0+ ο οποίος θεωρείται αρκετά ισχυρός. Η πλακέτα διαθέτει δύο κουμπιά τα οποία συμβολίζονται με τα γράμματα A και B. Επιπλέον υπάρχουν 25 led τα οποία είναι σε μορφή πλέγματος 5x5 και μέσω αυτών μπορούμε να απεικονίσουμε είτε γραφικά είτε κάποιο κείμενο. Το Micro:bit επίσης έχει επιταχυνσιόμετρο και μαγνητόμετρο, όπου του παρέχουν την δυνατότητα να ανιχνεύει την θέση του, αλλά και έναν αισθητήρα φωτός και ένα θερμόμετρο. Τέλος παρέχονται πέντε μεταλλικοί στρόγγυλοι ακροδέκτες όπου οι χρήστες μπορούν να τους χρησιμοποιήσουν τους αριθμούς από 0-2 ως είσοδο είτε ως έξοδο. Τον



ακροδέκτη 3V τον χρησιμοποιούν για τροφοδοσία και τον GND για την γείωση. Στους ακροδέκτες 0-2 έχουμε την δυνατότητα διασύνδεσης εξαρτημάτων όπως (Leds, αισθητήρες κ.α) με κύριο πλεονέκτημα τον έλεγχο από τον μικροελεγκτή.



Εικόνα 30 - Παράδειγμα Micro:bit στο Make a code

Ο προγραμματισμός του Micro:bit μπορεί να πραγματοποιηθεί με αρκετά προγραμματιστικά περιβάλλοντα (IDE, Integrated Development Environment). Αυτό που θα χρησιμοποιήσουμε εμείς είναι το Make a code, επειδή είναι αρκετά εύχρηστο και απλό σε μικρά παιδιά. Διαθέτει blocks προγραμματισμού και το κάνει οικείο στα μικρότερα παιδιά και για τα παιδιά τα οποία είναι προχωρημένα διαθέτει την γλώσσα προγραμματισμού Javascript. Επιπλέον μας παρέχει την ευκαιρία να χρησιμοποιήσουμε την πλατφόρμα του make a code διαδικτυακά αλλά και με εγκατάσταση στο προσωπικό μας υπολογιστή, με την προϋπόθεση ότι έχουμε μεταγλωττίσει τουλάχιστον ένα έργο.

Επίσης η πλατφόρμα του Make a code μας δίνει την ευκαιρία να προγραμματίσουμε το Micro:bit χωρίς να έχουμε τον μικροελεγκτή αλλά από τον προσομοιωτή τον οποίο διαθέτει. Παρακάτω θα υλοποιήσουμε ένα οδηγό εκμάθησης Micro:bit ενσωματωμένο σε φύλλα εργασίας κλιμακωτής δυσκολίας και στο τέλος του οδηγού θα πραγματοποιήσουμε δύο εκπαιδευτικά σενάρια.

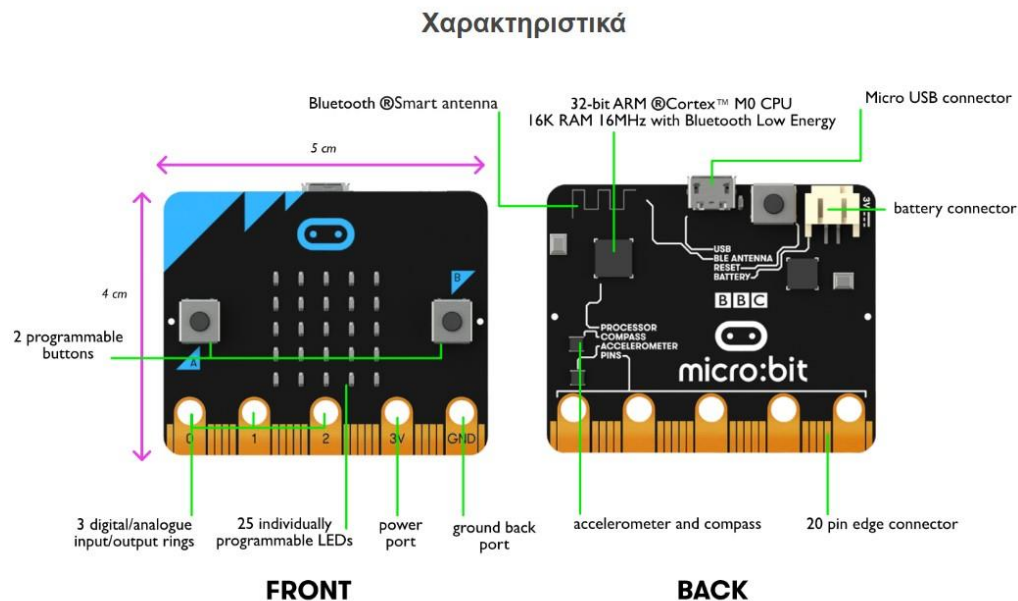
Κάθε φύλλο εργασίας θα περιέχει:

- ✚ Σύντομη περιγραφή του σεναρίου
- ✚ Τα εξαρτήματα τα οποία θα χρησιμοποιήσουμε



- ✚ Το διάγραμμα του κυκλώματος
- ✚ Τα αναλυτικά βήματα επίλυσης του φύλλου εργασίας
- ✚ Τον απαιτούμενο κώδικα για την επίλυση του φύλλου εργασίας (Blocks και Javascript).

Τα χαρακτηριστικά του Micro:bit

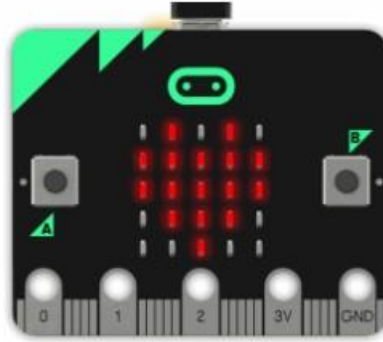


Εικόνα 31 - Τα εξαρτήματα του Micro:bit

- ✚ 25(5x5) ξεχωριστά προγραμματιζόμενα Leds.
- ✚ κουμπιά που υπάρχει η δυνατότητα προγραμματισμού.
- ✚ Pins όπου μπορούμε να κάνουμε φυσικές συνδέσεις.
- ✚ Αισθητήρα φωτός.
- ✚ Αισθητήρα θερμοκρασίας.
- ✚ Αισθητήρα κίνησης.
- ✚ Bluetooth.
- ✚ Ασύρματη επικοινωνία μέσω ραδιοκυμάτων.
- ✚ Θύρα διασύνδεσης Usb.

Τα Leds του Micro:bit

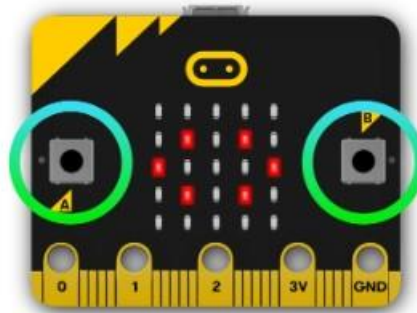
Οι δίοδοι εκπομπής φωτός (LEDs, Light emitting diodes) στις ημέρες μας χρησιμοποιούνται σε όλο και περισσότερες ηλεκτρονικές συσκευές. Το Micro:bit μας παρέχει 25 Leds με την δυνατότητα προγραμματισμού και εμφάνιση κειμένων, αριθμών και εικόνων.



Εικόνα 32 - Παράδειγμα LEDs στο Micro:bit

Τα κουμπιά του Micro:bit

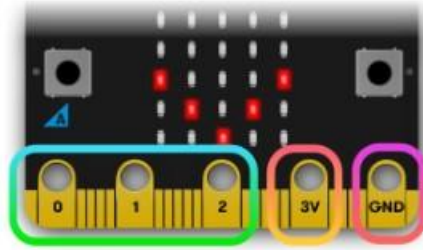
Το Micro:bit διαθέτει 2 προγραμματιζόμενα κουμπιά με τα γράμματα Α και Β αντίστοιχα. Βρίσκονται στο μπροστά μέρος της συσκευής και έχουμε την δυνατότητα να τσεκάρουμε πότε χρησιμοποιούνται, για να ελέγχουμε όταν εκτελείται κώδικας στην συσκευή.



Εικόνα 33 - Παράδειγμα κουμπιών στο Micro:bit

Τα Pins στο Micro:bit

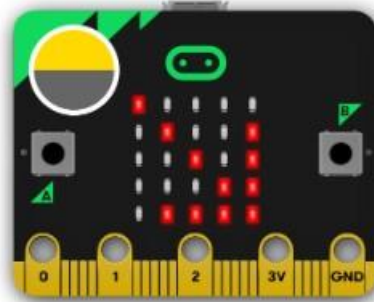
Το Micro:bit διαθέτει 5 εξωτερικούς ακροδέκτες στο κάτω μέρος της συσκευής, όπου μας παρέχει ενσύρματα την επικοινωνίας με τον εξωτερικό κόσμο. Κατά συνέπεια έχουμε την ευκαιρία να συνδέσουμε διάφορα εξαρτήματα όπως κινητήρες, LEDs, αισθητήρες κ.α.



Εικόνα 34 - Παράδειγμα των Pins στο Micro:bit

Ο αισθητήρας φωτός στο Micro:bit

Το Micro:bit μας παρέχει την ευκαιρία για αντιστροφή της λειτουργίας των LEDs και χρήση ως είσοδο ανιχνεύοντας την παρουσία φωτός.

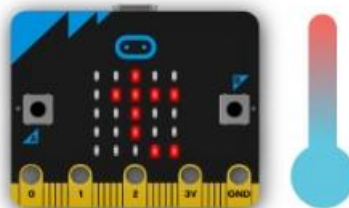


Εικόνα 35 - Παράδειγμα αισθητήρα φωτός στο Micro:bit

109

Ο αισθητήρας θερμοκρασίας στο Micro:bit

Το Micro:bit μας παρέχει έναν αισθητήρα θερμοκρασίας όπου ανιχνεύει την τρέχουσα θερμοκρασία σε βαθμούς κελσίου.

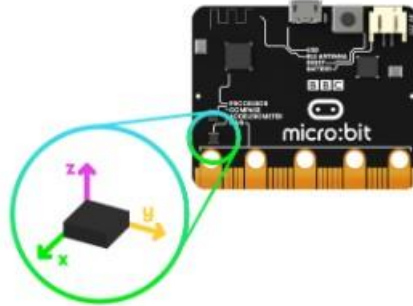


Εικόνα 36 - Παράδειγμα αισθητήρα θερμοκρασίας στο Micro:bit



Το επιταχυνσιόμετρο στο Micro:bit

Το Micro:bit διαθέτει ένα επιταχυνσιόμετρο με το οποίο μπορούμε να ανιχνεύσουμε πότε η συσκευή έχει μετακινηθεί.



Εικόνα 37 - Παράδειγμα του επιταχυνσιόμετρου στο Micro:bit

Η πυξίδα στο Micro:bit

Ο αισθητήρας της πυξίδας στο Micro:bit μας παρέχει την δυνατότητα ανίχνευσης του μαγνητικού πεδίου της γης. Αυτό έχει ως αποτέλεσμα με την βοήθεια της να μπορούμε να προσδιορίζουμε την κατεύθυνση της συσκευής. Να σημειωθεί βέβαια πως για να λειτουργήσει σωστά ο αισθητήρας θα πρέπει να καλιμπραριστεί πριν την χρήση.

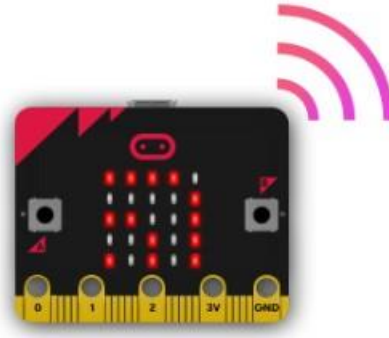
110



Εικόνα 38 - Παράδειγμα πυξίδας στο Micro:bit

Η ραδιοεπικοινωνία στο Micro:bit

Η συσκευή του Micro:bit παρέχει ασύρματη επικοινωνία μεταξύ άλλων συσκευών ή άλλες συσκευές Micro:bit μέσω των ραδιοκυμάτων. Κατά συνέπεια με την χρήση της υπάρχει η δυνατότητα δημιουργίας παιχνιδιών με άλλους χρήστες.



Εικόνα 39 - Παράδειγμα ραδιοεπικοινωνίας στο Micro:bit

Το Bluetooth στο Micro:bit

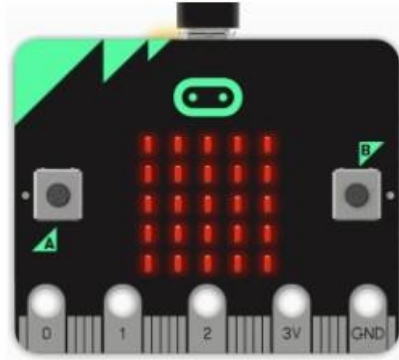
Η συσκευή του Micro:bit περιέχει μια κεραία BLE (Bluetooth Low Energy) μέσω της οποίας γίνεται η επικοινωνία. Έτσι η συσκευή μπορεί να συνδεθεί με κινητά τηλέφωνα, υπολογιστές, tablet κ.α. Αυτό δημιουργεί έναν έλεγχο μεταξύ των συσκευών και του Micro:bit καθώς και το αντίστροφο. Οι χρήστες έχουν την ευκαιρία αποστολής εντολών από το Micro:bit στις συσκευές και το ανάποδο. Η μόνη προϋπόθεση λειτουργίας του Bluetooth είναι η ζεύξη με την ανάλογη συσκευή.



Εικόνα 40 - Παράδειγμα Bluetooth στο Micro:bit

Η θύρα USB του Micro:bit

Η συσκευή του Micro:bit πραγματοποιεί σύνδεση με τον Η/Υ με την βοήθεια μιας θύρας Micro USB. Με την χρήση της έχουμε την δυνατότητα μεταφόρτωσης των προγραμμάτων που έχουμε δημιουργήσει για το Micro:bit, αλλά παρέχουμε και τροφοδοσία στην συσκευή.



Εικόνα 41 - Παράδειγμα θύρας Micro-USB στο Micro:bit

4.3 Οδηγός εκμάθησης Micro:bit με φύλλα εργασίας

4.3.1 Φύλλο εργασίας 1 (Blink a led)

Σ' αυτό το φύλλο εργασίας θα προσπαθήσουμε να προγραμματίσουμε το Micro:bit έτσι ώστε να αναβοσβήνει ένα φωτάκι της οθόνης του. Το μέγεθος της οθόνης είναι 5x5 led. Το συγκεκριμένο φύλλο εργασίας θα πραγματοποιηθεί στην πλατφόρμα του Make a code. Κατά συνέπεια έχουμε την δυνατότητα να εργαστούμε στο διαδίκτυο ή μόνο στον υπολογιστή μας έπειτα από την εγκατάσταση της πλατφόρμας, ανάλογα τις παροχές που έχουμε.

112

Στόχοι:

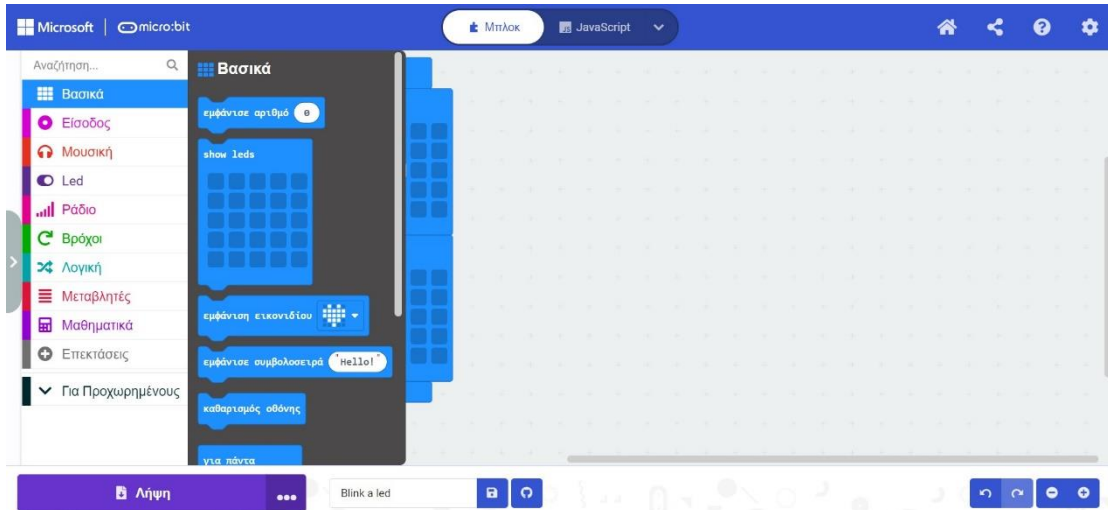
- ✚ Εξοικείωση και παιχνίδι στην πλατφόρμα προγραμματισμού του Make a code.
- ✚ Κατανόηση της βασικής δομής του προγράμματος από τους εκπαιδευόμενους.

Εξαρτήματα:

1. Πλακέτα Micro:bit.
2. Καλώδιο διασύνδεσης Micro-USB.

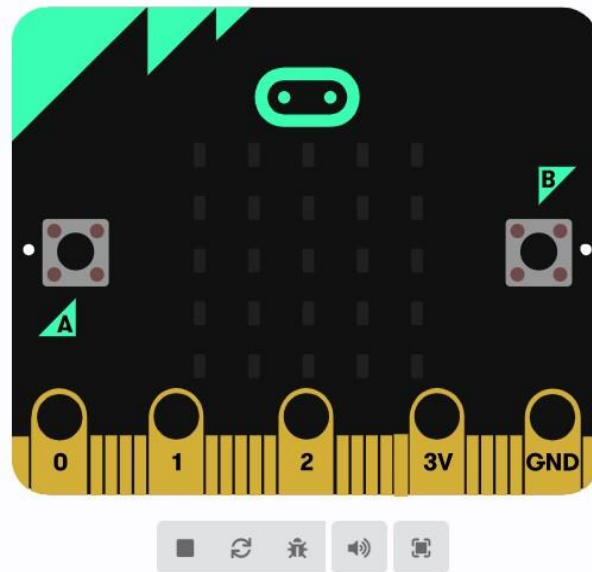
Βήματα του φύλλου εργασίας

1. Ανοίγουμε το πρόγραμμα του Make a code στον H/Y ή την σελίδα <https://makecode.microbit.org> σε έναν φυλλομετρητή.
2. Το πρόγραμμα Make a code έχει αρκετές επιλογές, όπου θα τις εξηγήουμε ανάλογα με το φύλο εργασίας το οποίο θα δουλεύουμε.



Αρχικά η κύρια οθόνη του Make a code χωρίζεται σε 5 ξεχωριστά κουμπιά. Το κουμπί με το σπιτάκι μας δίνει την δυνατότητα να ξεκινήσουμε ένα καινούργιο πρότζεκτ ή να πραγματοποιήσουμε αποθήκευση. Ακριβώς από δίπλα έχουμε το κουμπί της κοινοποίησης. Με το συγκεκριμένο κουμπί έχουμε την ευκαιρία να κοινοποιήσουμε το πρόγραμμα που έχουμε δημιουργήσει, έτσι ώστε να υπάρχει δυνατότητα χρήσης και από άλλους χρήστες. Στο πάνω μέρος της οθόνης στο κέντρο έχουμε ακόμη δύο κουμπιά όπου είναι υπεύθυνα για τον προγραμματισμό του Micro:bit. Στο κουμπί που έχει το όνομα μπλοκ μπορούμε να προγραμματίζουμε με την βοήθεια των πλακιδίων. Το μόνο που χρειάζεται είναι να τοποθετήσουμε στην σωστή σειρά τα πλακίδια, κάτω από τα πλακίδια κατά την έναρξη και το πλακίδιο για πάντα, και ο μικροελεγκτής θα υλοποιήσει αυτό που θέλουμε. Το δίπλα κουμπί που ονομάζεται Javascript μας παραπέμπει στο να γράψουμε κώδικα. Είναι για πιο προχωρημένους μαθητές και δεν θα το αναλύσουμε στα παρακάτω φύλλα εργασίας. Τέλος έχουμε ακόμη δύο κουμπιά πάνω δεξιά το ένα είναι ένα ερωτηματικό και το άλλο ένα γρανάζι. Το πρώτο μας παρέχει βοήθεια σχετικά με το Micro: bit και διάφορα παραδείγματα πάνω σ' αυτό. Ενώ το γρανάζι περιέχει όλες τις ρυθμίσεις που μας παρέχει η πλατφόρμα του Micro:bit.

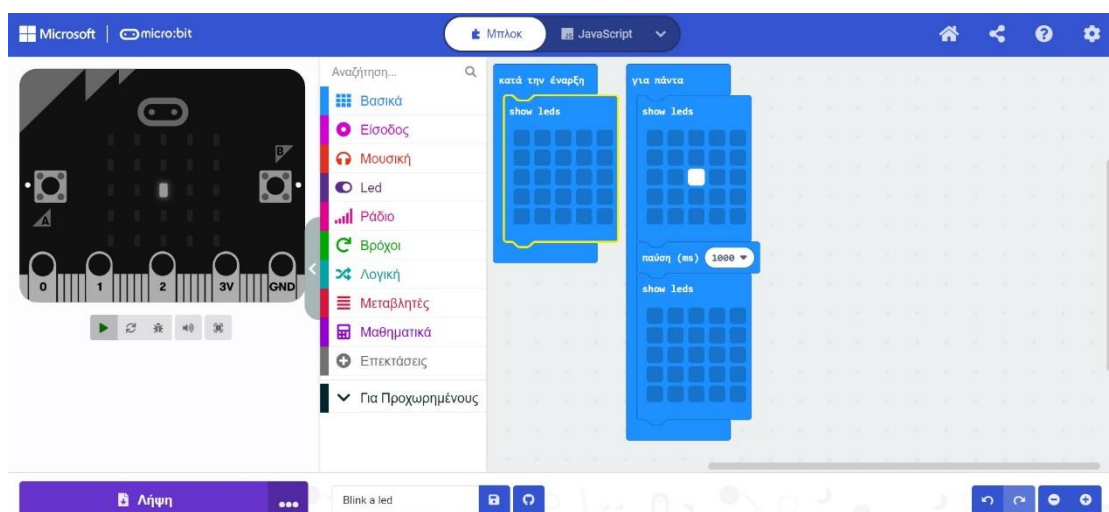
Πάνω αριστερά βρίσκεται ο προσομοιωτής όπου μας παρουσιάζει όποιο πρόγραμμα υλοποιούμε στα μπλοκς. Από κάτω έχει πέντε κουμπιά λειτουργίας του προσομοιωτή. Το πρώτο από τ αριστερά ξεκινάει και σταματάει τον προσομοιωτή. Το δεύτερο πραγματοποιεί επανεκκίνηση, ενώ το τρίτο κάνει την λειτουργία του προσομοιωτή πιο αργή. Το τέταρτο κουμπί είναι υπεύθυνο για τον ήχο του καθώς τον ανοίγει και τον κλείνει. Τέλος το πέμπτο και τελευταίο κουμπί πραγματοποιεί μεγιστοποίηση της οθόνης του προσομοιωτή.



Εικόνα 42 - Παράδειγμα προσομοιωτή Micro:bit

Υπάρχουν στα μπλοκ δύο συναρτήσεις τις οποίες θα χρησιμοποιούμε σ' όλα τα φύλλα εργασίας. Η πρώτη ονομάζεται “κατά την έναρξη” όπου όποια εντολή τοποθετήσουμε εκεί μέσα θα πραγματοποιηθεί με την έναρξη του προσομοιωτή μία φορά. Η δεύτερη ονομάζεται “Για πάντα” όπου όποια εντολή τοποθετήσουμε μέσα θα πραγματοποιείται συνέχεια. Κατά συνέπεια εμείς στο συγκεκριμένο φύλλο εργασίας θέλουμε να αναβοσβήνουμε ένα Led, για όσο είναι σε λειτουργία ο προσομοιωτής. Η παρακάτω εικόνα θα παρουσιάσει τα αναγκαία μπλοκ για να κάνει blink ο προσομοιωτής ένα συγκεκριμένο Led.

114



Εικόνα 43 - Παράδειγμα Blink a led στο Micro:bit



*Ενδιάμεσα στις δύο εντολές show leds χρησιμοποιούμε μία εντολή παύσης ενός δευτερολέπτου για να μπορούμε να βλέπουμε την διαφορά από την μία εντολή στην άλλη.

Κώδικας σε Javascript

```
basic.showLeds(`
  .....
  .....
  .....
  .....
  .....
  `)
basic.forever(function () {
  basic.showLeds(`
    .....
    .....
    ..#..
    .....
    .....
    `)
  basic.pause(1000)
  basic.showLeds(`
    .....
    .....
    .....
    .....
    .....
    `)
})
```



4.3.2 Φύλλο εργασίας 2 Το χαρούμενο και το λυπημένο

Emoji

Εισαγωγή: Σ' αυτό το φύλλο εργασίας θα πραγματοποιήσουμε την πρώτη μας δραστηριότητα η οποία θα περιλαμβάνει γραφικό περιβάλλον. Κατά συνέπεια στην πλατφόρμα του Make a Code όταν την ολοκληρώσουμε θα εμφανίζονται χαρούμενα και λυπημένα Emoji.

Στόχοι:

Αφού ολοκληρώσουμε το φύλλο εργασίας θα έχουμε την δυνατότητα να :

- ✚ Ανοίγουμε και να χειριζόμαστε τις βασικές λειτουργίες του προγραμματιστικού περιβάλλοντος του Make a Code.
- ✚ Αλλάζουμε την γλώσσα λειτουργίας της πλατφόρμας.
- ✚ Χειριζόμαστε τα Led της οθόνης του Micro:bit.
- ✚ Δημιουργούμε απλά προγράμματα με τα πλακίδια της εμφάνιση εικονιδίου και της παύσης.

Απαιτούμενα υλικά:

Η/Υ, Διασύνδεση στο διαδίκτυο.

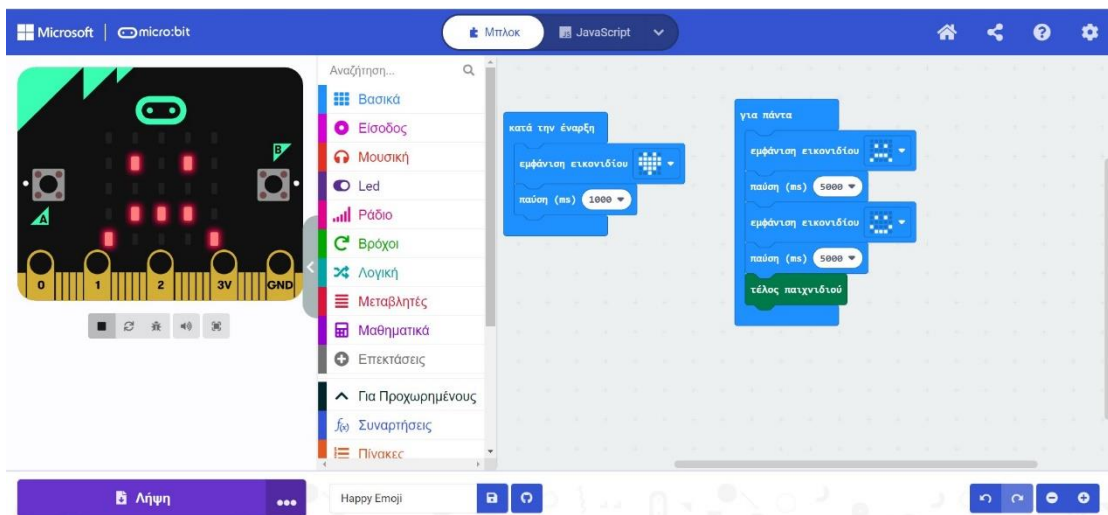
Βήματα φύλλου εργασίας

1. Ανοίγουμε έναν Browser στον Η/Υ και μπαίνουμε στην σελίδα :
<https://makecode.microbit.org/>
2. Από την κατηγορία βασικά παίρνουμε το πλακίδιο εμφάνιση εικονιδίου, επιλέγουμε το σχήμα με την καρδούλα και το τοποθετούμε στο πλακίδιο που λέει κατά την έναρξη. Εν συνεχεία παίρνουμε από την κατηγορία βασικά μία παύση και την προσαρμόζουμε στην ώρα που θέλουμε να φαίνεται η καρδιά κατά την έναρξη. Οτιδήποτε τοποθετήσουμε στο πλακίδιο κατά την έναρξη θα συμβεί μόνο μία φορά κατά την εκκίνηση του προσομοιωτή.
3. Αφού ολοκληρώσουμε τις τοποθετήσεις αυτές από την καρτέλα βασικά παίρνουμε ξανά το πλακίδιο εμφάνιση εικονιδίου, αλλά αυτή την φορά του δίνουμε το σχήμα μιας λυπημένης φάτσας. Επιπλέον το τοποθετούμε κάτω από το πλακίδιο για πάντα και από κάτω βάζουμε ένα πλακίδιο παύσης. Ο λόγος είναι για να μπορούμε να παρατηρήσουμε την αλλαγή από το λυπημένο



το Emoji σε χαρούμενο. Σε διαφορετική περίπτωση η κίνηση δεν θα ήταν ομαλή και δεν θα μπορούσαμε να παρατηρήσουμε την εναλλαγή.

4. Τέλος κάτω από την παύση τοποθετούμε ξανά ένα πλακίδιο με το όνομα εμφάνιση εικονιδίου, όπου αυτή την φορά έχει μέσα στην χαρούμενη φάτσα, και από κάτω τοποθετούμε ακόμη ένα πλακίδιο παύσης για όσα δευτερόλεπτα θέλουμε. Στην περίπτωση που αφήσουμε έτσι τον κώδικα το Micro: bit θα εναλλάσσει τις φάτσες από λυπημένη σε χαρούμενη για όσο χρόνο είναι σε λειτουργία. Οπότε για να ολοκληρωθεί το παιχνίδι από την καρτέλα παιχνίδι παίρνουμε το πλακίδιο τέλος παιχνιδιού, το τοποθετούμε στο τέλος μετά την παύση και μετά την εναλλαγή μας εμφανίζει στην οθόνη του Micro: bit Game Over.



117

Εικόνα 44 - Παράδειγμα Sad n Happy Emoji στο Micro: bit

Κώδικας σε Javascript

```
basic.showIcon(IconNames.Heart)
basic.pause(1000)
basic.forever(function () {
  basic.showIcon(IconNames.Sad)
  basic.pause(5000)
  basic.showIcon(IconNames.Happy)
  basic.pause(5000)
  game.gameOver()
})
```



Επιπλέον δραστηριότητες

Πραγματοποιούμε περισσότερες εναλλαγές σε Emoji για μεγαλύτερη εξοικείωση.

4.3.3 Φύλλο εργασίας 3 Εναλλαγή τυχαίων αριθμών

Εισαγωγή: Στο συγκεκριμένο φύλλο εργασίας θα χρησιμοποιήσουμε το Micro: bit για εναλλαγή τυχαίων αριθμών στην οθόνη.

Στόχοι:

Με την ολοκλήρωση του φύλλου εργασίας θα μπορούμε να :

- ✚ Εμφανίζουμε τυχαίους αριθμούς στην οθόνη του Micro: bit.
- ✚ Κάνουμε χρήση το πλακίδιο επανάληψης do για να ρυθμίζουμε τις επαναλήψεις το τυχαίων αριθμών.

118

Απαιτούμενα υλικά:

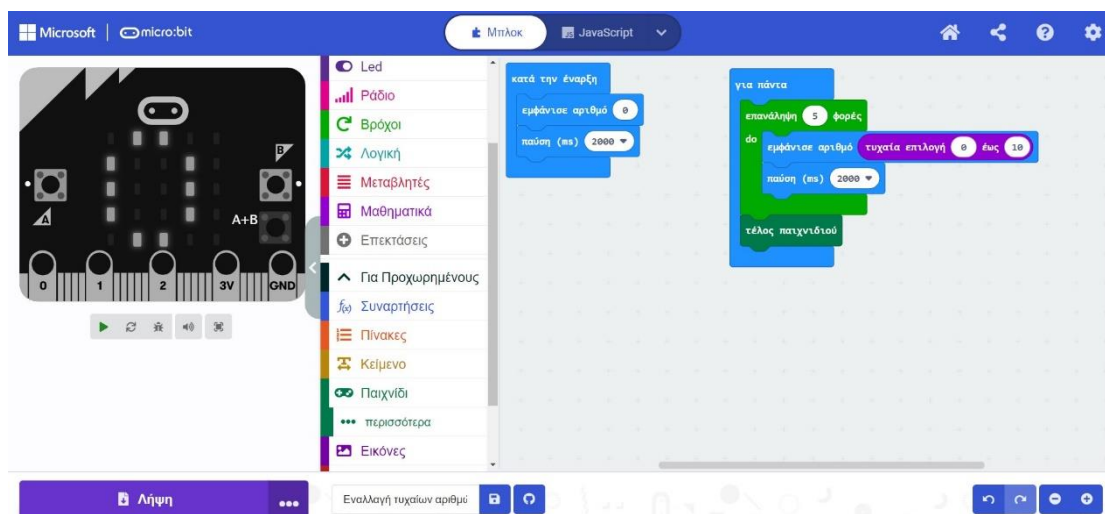
Χρήση του Η/Υ, Διασύνδεση στο διαδίκτυο.

Βήματα φύλλου εργασίας:

1. Ανοίγουμε από έναν φυλλομετρητή την ιστοσελίδα του <https://makecode.microbit.org/>.
2. Θα κάνουμε χρήση ξανά των συναρτήσεων κατά την έναρξη και την για πάντα.
3. Στην συνάρτηση κατά την έναρξη τοποθετούμε το πλακίδιο εμφάνισε αριθμό και επιλέγουμε τον αριθμό που θέλουμε να εμφανίζεται κατά την έναρξη του Micro: bit. Παράλληλα από κάτω τοποθετούμε ένα πλακίδιο παύσης για να κάνουμε την λειτουργία του Micro: bit πιο ομαλή.
4. Στην συνάρτηση για πάντα τοποθετούμε το πλακίδιο επανάληψη Do. Με την χρήση του συγκεκριμένου πλακιδίου εξασφαλίζουμε τις επαναλήψεις που θέλουμε πραγματοποιεί το Micro: bit στους τυχαίους αριθμούς. Το συγκεκριμένο πλακίδιο βρίσκεται στην κατηγορία των βρόγχων.



- Μέσα στο πλακίδιο της επανάληψης τοποθετούμε εμφάνισε αριθμό. Από την καρτέλα μαθηματικά επιλέγουμε την εντολή τυχαία επιλογή και τοποθετούμε το πλακίδιο μέσα στον αριθμό του πλακιδίου εμφάνισε αριθμό. Με τον τρόπο αυτό ο χρήστης έχει την δυνατότητα να επιλέξει το εύρος των αριθμών που θέλει να εμφανίζονται τυχαία μέσα στην οθόνη.
- Εν συνεχεία κάτω από την τυχαία επιλογή τοποθετούμε ένα πλακίδιο παύσης με σκοπό οι αλλαγές των αριθμών στην οθόνη του Micro: bit να πραγματοποιούνται ομαλά.
- Τέλος έξω από την εντολή επανάληψης Do αλλά μέσα στο για πάντα τοποθετούμε το πλακίδιο τέλος παιχνιδιού, όπου βρίσκετε στην καρτέλα παιχνίδι με σκοπό όταν ολοκληρωθούν οι επαναλήψεις που θέλουμε να ολοκληρωθεί το παιχνίδι.



Εικόνα 45 - Παράδειγμα εναλλαγής τυχαίων αριθμών στο Micro:bit

Κώδικας σε JavaScript

```
basic.showNumber(0)
basic.pause(2000)
basic.forever(function () {
  for (let index = 0; index < 5; index++) {
    basic.showNumber(randint(0, 10))
    basic.pause(2000)
  }
}
game.gameOver()
```



}}

Επιπλέον δραστηριότητες

- ✚ Αλλάξτε το εύρος των αριθμών για να δείτε τι συμβαίνει.
- ✚ Προσπαθήστε οι αριθμοί που επιλέξατε να εμφανίζονται διαδοχικά με την σωστή σειρά.

4.3.4 Φύλλο εργασίας 4 Εμφάνιση μηνυμάτων στην οθόνη του Micro:bit

Εισαγωγή: Σ' αυτό το φύλλο εργασίας θα χρησιμοποιήσουμε τα κουμπιά που διαθέτει το Micro:bit. Κατά την εκκίνηση του προγράμματος το Micro:bit θα μας εμφανίζει ένα χαμογελαστό Emoji. Εν συνεχεία θα μας εμφανίζει ένα μήνυμα για το πως μας λένε και ανάλογα με το κουμπί που θα χρησιμοποιούμε θα μας δίνει την ανάλογη απάντηση. Όλες οι απαντήσεις θα είναι σε συμβολοσειρές και θα εμφανίζονται ξεχωριστά κάθε γράμμα στην οθόνη του Micro:bit.

120

Στόχοι:

Με την ολοκλήρωση του φύλλου εργασίας θα έχουμε την δυνατότητα να:

- ✚ Εμφανίζουμε συμβολοσειρές στην οθόνη του Micro:bit.
- ✚ Μάθουμε την χρήση των κουμπιών που διαθέτει το Micro:bit.
- ✚ Να κατανοήσουμε την λειτουργία του πλακιδίου εάν τότε.

Απαιτούμενα υλικά:

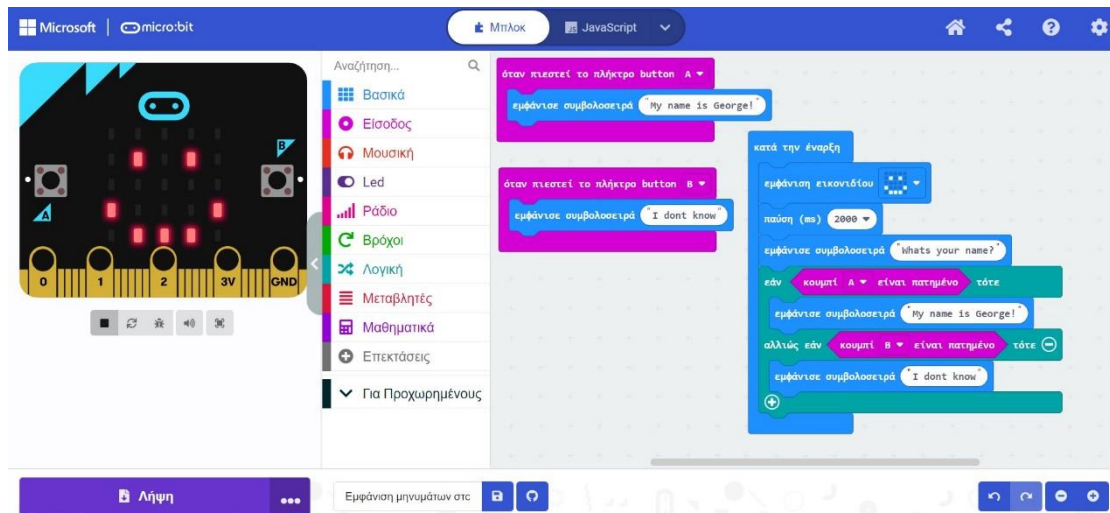
Χρήση του Η/Υ, διασύνδεση στο διαδίκτυο.

Βήματα φύλλου εργασίας:

1. Ανοίγουμε από έναν φυλλομετρητή την ιστοσελίδα του <https://makecode.microbit.org/>.
2. Στο συγκεκριμένο φύλλο εργασίας θα δουλέψουμε μόνο την συνάρτηση κατά την έναρξη, διότι η διαδικασία δεν θέλουμε να επαναλαμβάνεται. Κατά συνέπεια από την καρτέλα είσοδο παίρνουμε δύο πλακίδια τα οποία ονομάζονται όταν πιεστεί το πλήκτρο button.



3. Από την καρτέλα βασικά παίρνουμε δύο πλακίδια τα οποία ονομάζονται εμφάνιση συμβολοσειρά. Τοποθετούμε τα μηνύματα τα οποία θέλουμε και στο ένα πλακίδιο επιλέγουμε το button A και στο άλλο το B. Ο σκοπός μας είναι κατά την έναρξη του Micro:bit επιλέγοντας ένα από τα δύο κουμπιά να μας εμφανίζει το αντίστοιχο μήνυμα.
4. Στην συνάρτηση κατά την έναρξη αρχικά τοποθετούμε ένα πλακίδιο να εμφανίζει ένα χαρούμενο Εμοji και από κάτω ένα πλακίδιο παύσης.
5. Έπειτα επιλέγουμε ένα πλακίδιο για την εμφάνιση συμβολοσειράς από την καρτέλα βασικά. Τα μηνύματα εμφανίζονται στην οθόνη ανά γράμμα ξεχωριστά.
6. Εν συνεχεία από την καρτέλα λογική επιλέγουμε το πλακίδιο εάν τότε αλλιώς εάν. Στον προγραμματισμό η συγκεκριμένη εντολή θεωρείται δομή επιλογής. Έτσι δίνει την ευκαιρία στον χρήστη να επιλέξει το κουμπί του Micro:bit το οποίο θέλει. Μέσα στην συνθήκη του εάν τοποθετούμε το πλακίδιο που έχουμε πάρει από την κατηγορία είσοδος. Το πλακίδιο ονομάζεται κουμπί A είναι πατημένο. Με το τρόπο αυτό εξασφαλίζουμε πως αν ο χρήστης πιέσει το κουμπί A του Micro:bit, θα εμφανίσει την συμβολοσειρά την οποία θέλουμε.
7. Οπότε κάτω από το εάν τοποθετούμε το πλακίδιο εισαγωγή συμβολοσειράς και βάζουμε το μήνυμα το οποίο θέλουμε.
8. Την ίδια ακριβώς διαδικασία ακολουθούμε και για το ένα αλλιώς με την μόνη διαφορά πως στην συνθήκη το πλακίδιο που τοποθετείτε αφορά το κουμπί B του Micro:bit.
9. Τέλος κάτω από το εάν αλλιώς τοποθετείτε από την καρτέλα βασικά ξανά το πλακίδιο εμφάνιση συμβολοσειράς, με σκοπό να επιλέξουμε το μήνυμα που θέλουμε όταν πατάμε το κουμπί B του Micro:bit



Εικόνα 46 - Παράδειγμα εμφάνισης μηνυμάτων και συμβολοσειρών στο Micro:bit

Κώδικας σε JavaScript

```
input.onButtonPressed(Button.A, function () {
    basic.showString("My name is George!")
})
input.onButtonPressed(Button.B, function () {
    basic.showString("I dont know")
})
basic.showIcon(IconNames.Happy)
basic.pause(2000)
basic.showString("Whats your name?")
if (input.buttonIsPressed(Button.A)) {
    basic.showString("My name is George!")
} else if (input.buttonIsPressed(Button.B)) {
    basic.showString("I dont know")
}
```

122

Επιπλέον δραστηριότητες

- 🔧 Προσθέστε επιπλέον μηνύματα κατά την χρήση των κουμπιών του Micro:bit.



- ✚ Δοκιμάστε στην πολλαπλή δομή επιλογής (εάν..τότε, αλλιώς..εάν, αλλιώς..εάν) για να βάλετε ακόμη μία συμβολοσειρά, όταν πιέζονται ταυτόχρονα το A και B κουμπί του Micro:bit.

4.3.5 Φύλλο εργασίας 5 Παιδικό τραγούδι με την βοήθεια του Micro:bit

Σ' αυτό το φύλλο εργασίας θα πραγματοποιήσουμε ένα παιδικό τραγουδάκι με την βοήθεια του Micro:bit. Στο συγκεκριμένο φύλλο δεν είναι αναγκαίο να χρησιμοποιήσουμε την συνάρτηση κατά την έναρξη, ούτε και την για πάντα. Αυτό που θα χρειαστούμε είναι το πλακίδιο από το μενού είσοδος. Κατά την έναρξη της συσκευής πατώντας το κουμπί A το Micro:bit θα παίζει κανονικά ένα παιδικό τραγούδι. Όταν πατάμε το κουμπί B με μια συμβολοσειρά το Micro:bit θα μας παραπέμπει να πατήσουμε το κουμπί A για να ακούσουμε το τραγούδι.

Στόχοι:

Με την ολοκλήρωση του συγκεκριμένου φύλλου εργασίας θα πρέπει να:

- ✚ Χειριζόμαστε το μενού μουσική για να μπορέσουμε να υλοποιήσουμε το παιδικό τραγούδι.
- ✚ Μάθουμε πως λειτουργεί η επανάληψη do για να γλιτώσουμε πλακίδια κατά την σύνθεση του τραγουδιού.

Απαιτούμενα υλικά:

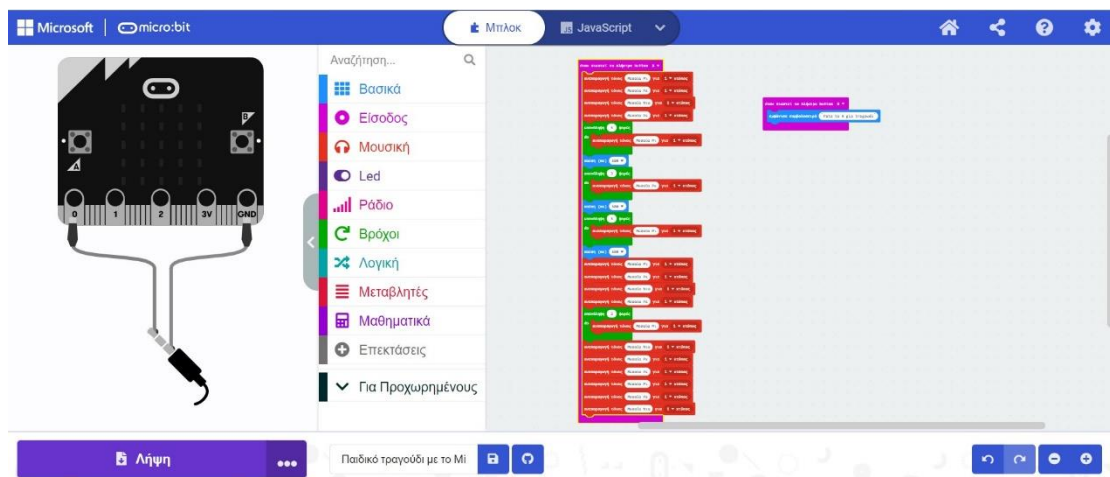
Χρήση του Η/Υ, Διασύνδεση στο διαδίκτυο.

Βήματα φύλλου εργασίας:

1. Ανοίγουμε από έναν φυλλομετρητή την ιστοσελίδα του <https://makecode.microbit.org/>.
2. Από το μενού είσοδο παίρνουμε 2 πλακίδια που λένε όταν πιεστεί το πλήκτρο A και όταν πιεστεί το πλήκτρο B.
3. Μέσα στο πλακίδιο όταν πιεστεί το πλήκτρο B, από το μενού βασικά παίρνουμε ένα πλακίδιο που ονομάζεται εμφάνισε συμβολοσειρά. Τοποθετούμε το μήνυμα που μας ενδιαφέρει δηλαδή να μας παραπέμπει να πατήσουμε το πλήκτρο A.



4. Μέσα στο πλακίδιο όταν πιεστεί το πλήκτρο A, από το μενού μουσική επιλέγουμε το πλακίδιο που λέει αναπαγωγή τόνος και επιλέγουμε την νότα που μας ενδιαφέρει.
5. Τοποθετούμε τις νότες στην σειρά την μία κάτω από την άλλη ωστόσο σε κάποιες νότες είναι αναγκαίο να τις επαναλάβουμε δύο και τρεις φορές. Έτσι από το μενού βρόχοι επιλέγουμε το πλακίδιο επανάληψης Do. Τοποθετώντας την νότα μέσα στην Do επιλέγουμε πόσες φορές θα γίνει επανάληψη η συγκεκριμένη νότα.
6. Επιπλέον ενδιάμεσα από κάθε νότα όταν είναι αναγκαίο από το μενού βασικά, παίρνουμε πλακίδια παύσης και τα προσαρμόζουμε στις ανάγκες του τραγουδιού. Το ίδιο μπορεί να συμβεί και έπειτα από πλακίδιο επανάληψης Do.



Εικόνα 47 - Παράδειγμα παιδικού τραγουδιού στο Micro:bit

Κώδικας σε JavaScript

```
input.onButtonPressed(Button.A, function () {  
  music.playTone(330, music.beat(BeatFraction.Whole))  
  music.playTone(294, music.beat(BeatFraction.Whole))  
  music.playTone(262, music.beat(BeatFraction.Whole))  
  music.playTone(294, music.beat(BeatFraction.Whole))  
  for (let index = 0; index < 3; index++) {  
    music.playTone(330, music.beat(BeatFraction.Whole))  
  }  
})
```



```
basic.pause(500)
for (let index = 0; index < 3; index++) {
  music.playTone(294, music.beat(BeatFraction.Whole))
}
basic.pause(500)
for (let index = 0; index < 3; index++) {
  music.playTone(330, music.beat(BeatFraction.Whole))
}
basic.pause(500)
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
for (let index = 0; index < 3; index++) {
  music.playTone(330, music.beat(BeatFraction.Whole))
}
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
})
input.onButtonPressed(Button.B, function () {
  basic.showString("Pata to A gia tragoudi")
})
```

Επιπλέον δραστηριότητες

- ✚ Στο κουμπί A χρησιμοποιήστε ένα διαφορετικό τραγούδι, χρησιμοποιώντας με διαφορετικό τρόπο τα πλακίδια των επαναλήψεων αλλά και των παύσεων.
- ✚ Στο κουμπί B τοποθετήστε ένα νέο τραγούδι, το οποίο θα παίζει όταν πατάμε το συγκεκριμένο πλήκτρο.



5. Συμπεράσματα

5.1 Επίλογος

Τα STEM και η εκπαιδευτική ρομποτική στις μέρες μας έχουν προσελκύσει τον ενδιαφέρον μαθητών και εκπαιδευτικών. Θεωρείται ένα χρήσιμο εργαλείο το οποίο εισάγει τους μαθητές ομαλά στον κόσμο του προγραμματισμού και της ρομποτικής. Η σύγχρονη εκπαίδευση έχει ως αναπόσπαστο κομμάτι την εκπαιδευτική ρομποτική, καθώς θεωρείται πολύπλευρη και είναι προσιτή και διασκεδαστική για εκπαιδευτικούς και μαθητές. Το εργαλείο εκμάθησης που χρησιμοποιεί αγγίζει αρκετούς τομείς όπως η επιστήμη, η τεχνολογία, η μηχανική και τα μαθηματικά. Επιπλέον βελτιώνει την ομαδοσυνεργατική μάθησης καθώς οι εκπαιδευόμενοι τις περισσότερες φορές εργάζονται σε ομάδες, αναπτύσσει τις δεξιότητες τους και την επικοινωνία. Τέλος η εκπαιδευτική ρομποτική δημιουργεί ένα ευχάριστο και δημιουργικό μαθησιακό κλίμα και κάνει τους εκπαιδευόμενους, να έχουν μεγαλύτερη ενασχόληση με όλα τα αντικείμενα της.

Κατά συνέπεια θα πρέπει να ενσωματωθεί το συντομότερο σε όλα τα σχολεία αλλά και στα πανεπιστήμια. Διότι όπως βλέπουμε και σ' αυτό τον οδηγό εκμάθησης η δυσκολία οδηγείται κλιμακωτά και μπορεί να φτάσει σε επίπεδα πανεπιστημίου, όσο ν' αφορά τις απαιτούμενες γνώσεις. Είναι αναγκαίο λοιπόν η εκπαιδευτική ρομποτική να θεωρηθεί σημείο αναφοράς σε όλες τις εκπαιδευτικές βαθμίδες, γιατί εξασφαλίζει σε όσους ασχολούνται μ' αυτή επιτυχία σε τεχνολογικά ζητήματα και όχι μόνο, γνώσεις και παράλληλα διασκέδαση.

5.2 Συμπεράσματα

Ο κόσμος της ρομποτικής πλέον είναι γεγονός. Άθελα μας ή εν γνώση μας όλοι έχουμε κάποια ενασχόληση με το αντικείμενο. Στην συγκεκριμένη διπλωματική εργασία ασχολήθηκα με την σύνταξη οδηγιών εκμάθησης Micro:bit και Arduino.

Εκτός από τα φύλλα εργασίας τα οποία έστησα στον οδηγό εκμάθησης Arduino συμπεριέλαβα δύο διδακτικά σενάρια, τα οποία η δυσκολία τους είναι για προχωρημένους μαθητές της τελευταίας τάξης του λυκείου. Προσπάθησα η δυσκολία



των φύλλων εργασίας να πηγαίνει κλιμακωτά και να ξεκινάει από το Micro:bit, για τις τελευταίες τάξεις του δημοτικού και τις πρώτες δύο του γυμνασίου. Έπειτα να συνεχίζει με το Arduino στην τελευταία τάξη του γυμνασίου και να περιέχει όλες τις τάξεις του λυκείου μέχρι την αποφοίτηση των μαθητών. Τα διδακτικά σενάρια επιλέγονται από τον εκπαιδευτικό διότι είναι για προχωρημένους μαθητές.

Επέλεξα την πλατφόρμα του Micro:bit για τις μικρότερες τάξεις γιατί περιέχει λιγότερες εισόδους/ εξόδους και είναι πιο ελκυστικό σε μικρά παιδιά. Θεωρώ πως κάνει μία πιο ομαλή εισαγωγή στον προγραμματισμό και την εκπαιδευτική ρομποτική. Έπειτα ακολουθεί το Arduino το οποίο είναι για μεγαλύτερα παιδιά και απαιτούνται εκτός από γνώσεις προγραμματισμού, γνώσεις ηλεκτρονικής και περιέχει πολυπλοκότερες συνδεσμολογίες. Σίγουρα και στις δύο πλατφόρμες μπορούμε να εμβαθύνουμε ως προς την δυσκολία των έργων και να στηρίξουν και τριτοβάθμια εκπαίδευση, ωστόσο η συγκεκριμένη διπλωματική υλοποιήθηκε για τις ανάγκες της πρωτοβάθμιας αλλά και της δευτεροβάθμιας εκπαίδευσης.

Έρευνα και προσωπική εμπειρία

Η συγκεκριμένη διπλωματική εργασία συντάχθηκε διότι υπήρχε η δυνατότητα της άμεσης δοκιμής των μικροελεγκτών στις βαθμίδες εκπαίδευσης, στις οποίες αναφέρεται. Κατά συνέπεια ως αναπληρωτής εκπαιδευτικός πληροφορικής πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, πραγματοποιήθηκαν διάφορες δοκιμές στην πρωτοβάθμια και στην δευτεροβάθμια εκπαίδευση.

Την περσινή χρονιά επειδή εργαζόμουν στην πρωτοβάθμια εκπαίδευση δοκίμασα να ενσωματώσω ομαλά τον μικροελεγκτή του Micro:bit, στις τάξεις της πέμπτης και έκτης δημοτικού. Τα παιδιά συνάντησαν αρκετά προβλήματα κατανόησης της πλατφόρμας καθώς και στον προγραμματισμό με τα πλακίδια. Έτσι επέλεξα να ενσωματώσω πρώτα το scratch Jr το οποίο δεν έχει καθόλου συνδεσμολογίες, παρά μόνο προγραμματισμό πλακιδίων. Τα παιδιά ξεκίνησαν να ανταποκρίνονται και να καταλαβαίνουν κάποια πράγματα πάνω στον προγραμματισμό πλακιδίων.

Εργαστήκαμε πάνω στο Scratch Jr τουλάχιστον δύο μήνες κάνοντας αρκετά φύλλα εργασίας και παραδείγματα. Έπειτα αποπειράθηκα ξανά να ενσωματώσω στις ίδιες τάξεις το Micro:bit στην πλατφόρμα του Make a code.

Με μεγάλη μου χαρά είδα πως τα παιδιά ξεκίνησαν να ανταποκρίνονται στον μικροελεγκτή με μία μικρή δυσκολία στις συνδεσμολογίες. Έτσι μαζί με την βοήθεια



του Geompris άλλη παιδαγωγική πλατφόρμα η οποία περιέχει εκμάθηση απλών ηλεκτρονικών συνδεσμολογιών, υλοποιήσαμε αυτά τα πέντε φύλλα εργασίας που σας παρουσιάζω σ' αυτή την διπλωματική. Δυστυχώς δεν είχαμε την δυνατότητα να περάσουμε τα φύλλα εργασίας στον μικροελεγκτή διότι στο σχολείο δεν διέθετε τον εν λόγω ελεγκτή, ωστόσο όλοι οι μαθητές έδειξαν μεγάλο ενδιαφέρον ακόμη και με την πλατφόρμα του Make a code.

Την χρονιά που ήμουν τοποθετημένος στην δευτεροβάθμια εκπαίδευση σε γυμνάσιο αλλά και σε λύκειο επιχείρησα να ενσωματώσω το Arduino στο μάθημα μου. Αφού πραγματοποιήσαμε μια μικρή εισαγωγή στο Scratch για να αντιληφθώ το επίπεδο των μαθητών στον προγραμματισμό πλακιδίων, έπειτα θεώρησα σωστό να επαναλάβω την εισαγωγή και τα φύλλα εργασίας στο Micro:bit. Οι μαθητές του γυμνασίου ανταποκρίθηκαν στον Micro:bit χωρίς ωστόσο μεγάλη επιτυχία, καθώς οι συνδεσμολογίες παρότι απλές έμοιαζαν να τους δυσκολεύουν. Έτσι πραγματοποίησα μόνο μία μικρή εισαγωγή στο Arduino στην Γ' Γυμνασίου και υλοποίησα τα πρώτα τρία φύλλα της συγκεκριμένης διπλωματικής.

Παράλληλα την ίδια χρονιά ήμουν τοποθετημένος και σε λύκειο και είχα την Α και Β λυκείου. Στην Α λυκείου αφού κάναμε το πρώτο εξάμηνο το App Inventor ενσωμάτωσα σταδιακά το Arduino. Οι μαθητές ανταποκρίθηκαν ως προς το κομμάτι του προγραμματισμού, αλλά προς το τέλος των φύλλων εργασίας προβληματίστηκαν και δυσκολεύτηκαν ως προς τις συνδεσμολογίες. Στο ίδιο μοτίβο κινήθηκαν και οι μαθητές της Β Λυκείου, όπου κατανόησαν καλύτερα και γρηγορότερα τα φύλλα εργασίας, ωστόσο παρουσίασαν τα ίδια προβλήματα στις συνδεσμολογίες.

Συμπερασματικά έπειτα από την μικρή έρευνα όπου πραγματοποίησα λόγο του επαγγέλματος μου, καταλήγω στο εξής συμπέρασμα. Η εισαγωγή των μαθητών στην εκπαιδευτική ρομποτική επιβάλλεται να πραγματοποιηθεί σταδιακά και τμηματικά.

Επειδή τα STEM εμπεριέχουν περισσότερες από μία ειδικότητες θα πρέπει ο εκπαιδευτής να δείξει μεγάλη προσοχή. Σύμμαχος του θα πρέπει να είναι η επανάληψη και η υπομονή. Επίσης προτείνω το Micro:bit να πραγματοποιείται στις τελευταίες τάξεις του δημοτικού έπειτα από καλή εκμάθηση του Scratch jr και να συνεχίζει στις υπόλοιπες τάξεις μέχρι την ολοκλήρωση του γυμνασίου. Επιπλέον να ακολουθεί το Arduino με μία πολύ καλή εισαγωγή στην Α' Λυκείου και να ολοκληρώνεται με το πέρας των σπουδών στο λύκειο του κάθε μαθητή.



5.3 Βιβλιογραφία

1. Πληροφορίες για το Arduino <https://www.arduino.cc/>
2. Ρομποτική: Φύλλα εργασίας για τις πλατφόρμες MICRO:BIT® ΚΑΙ ARDUINO®
3. Προγραμματισμός & Ρομποτική στο σχολείο και στο σπίτι με BBC micro:bit 2.
4. <https://makecode.microbit.org>
5. Διδακτική και σχεδιασμός εκπαιδευτικών δραστηριοτήτων STEM και ΤΠΕ - Σαράντος Ψυχάρης, Κωνσταντίνος Καλοβρέκτης
6. <http://ardublock.ru/avtor/002/en.html>
7. Ανάπτυξη εφαρμογών με το ARDUINO - 9789604189373 - Τζιόλα - Παπάζογλου Παναγιώτης.
8. <https://www.tinkercad.com/>
9. Ανάπτυξη Εφαρμογών με το Arduino, 2η Έκδοση Σπυρίδων - Πολυχρόνης Λιωνής Παπάζογλου Παναγιώτης.
10. Πουλάκης, Ε. (2015). Προγραμματίζοντας με τον μικροελεγκτή Arduino. Ε.Πουλάκης: Ηράκλειο.