



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΕΛΛΑΔΟΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ,
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO ΜΕΣΩ SCRATCH ΜΕ
ΤΗΝ ΧΡΗΣΗ ΕΦΑΡΜΟΓΩΝ WEB**

Πτυχιακή Εργασία του

Γιοβανίτσα Κωνσταντίνου (3861)

Επιβλέπων: Στ. Βολογιαννίδης, Επίκουρος Καθηγητής

ΣΕΡΡΕΣ, ΙΑΝΟΥΑΡΙΟΣ 2021

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

Περίληψη

Η συγκεκριμένη πτυχιακή εργασία, έχει ως θέμα τον Προγραμματισμό Arduino μέσω Scratch με την χρήση εφαρμογών στο διαδίκτυο. Αρχικά γίνεται μια εισαγωγή στην ηλεκτρονική πλακέτα Arduino. Θα εξεταστούν οι τρόποι προγραμματισμού του, με παραδείγματα και εικόνες των block γραφικών για την δημιουργία κώδικα. Μια μέθοδος με γραφικά στοιχεία, απλά και κατανοητά, που στόχο έχει την διασκεδαστική και γρήγορη εξοικείωση ενός χρήστη με την σύνθεση κώδικα του Arduino.

Για την εκτέλεση της εργασίας, θα χρησιμοποιηθεί ένα raspberry pi σε συνδυασμό με το Arduino Uno, μια σύνδεση στο internet και μία θήκη tablet ειδικά φτιαγμένη να μπορεί να φιλοξενεί το raspberry pi μαζί με μια οθόνη αφής 7 ιντσών. Μια πιο εύκολη εναλλακτική λύση, είναι η χρήση οποιουδήποτε monitor με ένα καλώδιο micro HDMI. Πέρα απο το Raspberry pi, μπορεί να χρησιμοποιηθεί οποιoδήποτε σύστημα με λειτουργικό Linux.

Στην σελίδα [GitHub](#), υπάρχει διαθέσιμο το open source λογισμικό BlocklyDuino για εγκατάσταση στο raspberry pi. Χωρίς ιδιαίτερη δυσκολία ενεργοποιούμε τον web server, όπου θα φιλοξενείται το περιβάλλον για την δημιουργία κώδικα με χρήση block γραφικών. Θα δούμε τα πηγαία αρχεία του λογισμικού που θα χρησιμοποιηθεί, έτσι ώστε να επιτευχθεί η συμβατότητα του στο σύστημα μας.

Στη συνέχεια , αφού θα έχει τελειοποιηθεί επιτυχώς η εγκατάσταση, θα είμαστε σε θέση να επισκεφτούμε απομακρυσμένα το περιβάλλον αυτό και να φτιάξουμε κώδικα τον οποίο, θα κάνουμε upload στο Arduino που συνδέεται σειριακά στο raspberry pi. Μέσα από παραδείγματα, θα δούμε βήμα-βήμα την διαδικασία προγραμματισμού στο BlocklyDuino

Η νέα μεταφρασμένη στα Ελληνικά, έκδοση του λογισμικού που θα χρησιμοποιηθεί διευκολύνει την εκμάθηση της γλώσσας προγραμματισμού Arduino μέσω scratch στην Ελλάδα. Η χρήση του web server είναι κατάλληλη για παρουσίαση του προγραμματισμού σε πολλούς ακροατές ταυτόχρονα, χρήσιμο για μια αίθουσα διδασκαλίας. Τα γραφικά στοιχεία εξυπηρετούν, από άποψη ευκολίας, την εξοικείωση με την γλώσσα κάθε αρχαρίου μαθητή, αλλά επίσης διευκολύνουν τον εκάστοτε

καθηγητή, να μπορέσει να μεταδώσει με απλό και κατανοητό τρόπο, το περιεχόμενο του μαθήματος του.

Λέξεις Κλειδιά:

Arduino, BlocklyDuino, Raspberry Pi, GitHub, Scratch, Local Server

Περιεχόμενα

Περίληψη	3
Περιεχόμενα.....	4
Πίνακας εικόνων	5
Εισαγωγή	7
1. Μικροελεγκτές-Μικροϋπολογιστές	9
1.1 Εισαγωγή στους μικροελεγκτές	9
1.1.1 Αναλυτικά τα πλεονεκτήματα των μικροελεγκτών είναι:	9
1.2 Η αρχιτεκτονική των μικροελεγκτών.....	11
1.2.1 Η Κεντρική Μονάδα Επεξεργασίας.....	11
1.2.2 Η μνήμη	12
1.2.3 Οι Μονάδες Εισόδου/Εξόδου (I/O)	12
1.3 Διαφορές Μικροελεγκτή-Μικροεπεξεργαστή.....	13
2. Το Arduino.....	14
2.1 Εισαγωγή στο Arduino	14
2.2 Τα είδη Arduino	15
Το Arduino UNO	15
Το Arduino Mega.....	16
Το Arduino Due	17
Το Arduino Ethernet REV 3	18
Το Arduino Nano	18
Το Arduino MKR Zero	19

2.3 Το λογισμικό	20
2.4 Το Scratch	22
3. Το Scratch για Arduino	24
3.1 Το S4A	24
3.2 Το Blockly.....	26
3.3 Το Blockly@rduino	27
4. Το BlocklyDuino	29
5. Το Raspberry Pi	30
5.1 Λήψη και εγκατάσταση του Arduino IDE.....	31
5.2 Λήψη και εγκατάσταση του BlocklyDuino	31
5.2.1 Επεξεργασία αρχείων.....	32
6. Προγραμματισμός σε BlocklyDuino	34
Παράδειγμα 1. Λειτουργία LED	34
Παράδειγμα 2. Αλλαγή φωτεινότητας LED (fading)	38
Παράδειγμα 3. Λειτουργία LED με ποτενσιόμετρο ως διακόπτης.....	41
Παράδειγμα 4. Έλεγχος σερβοκινητήρα με ποτενσιόμετρο	45
7. Συμπεράσματα	49
7.1 Το block factory	49
7.2 Συμβατά υπολογιστικά συστήματα.....	49
7.3 BlocklyDuino και σφάλματα	49
8. Βιβλιογραφία	50

Πίνακας εικόνων

Εικόνα 1. Δομή μικροϋπολογιστικού συστήματος	11
Εικόνα 2. Μια πρώιμη πλακέτα Arduino	14
Εικόνα 3. Το Arduino UNO	16

Εικόνα 4. Το Arduino Mega	16
Εικόνα 5. Το Arduino Due	17
Εικόνα 6. Το Arduino Ethernet REV 3	18
Εικόνα 7. Το Arduino Nano	19
Εικόνα 8. Το Arduino MKR Zero	19
Εικόνα 9. Το Arduino IDE	21
Εικόνα 10. Οι βιβλιοθήκες του IDE	22
Εικόνα 11. Το Scratch	22
Εικόνα 12. Το S4A	25
Εικόνα 13. Το Blockly	26
Εικόνα 14. Το Blockly@rduino	27
Εικόνα 15. Το BlocklyDuino	29
Εικόνα 16. Το Raspberry Pi imager	30
Εικόνα 17. Στιγμιότυπο από το αρχείο web_server.py	32
Εικόνα 18. Στιγμιότυπο από το αρχείο blockly-helper.js	33
Εικόνα 19. Στιγμιότυπο από το αρχείο hosts μέσω terminal	33
Εικόνα 20. Λειτουργία LED μέσω button	35
Εικόνα 21. Block και κώδικας παραδείγματος 1	36
Εικόνα 22. Αλλαγή φωτεινότητας LED	39
Εικόνα 23. Block και κώδικας παραδείγματος 2	40
Εικόνα 24. Αλλαγή φωτεινότητας LED με ποτενσιόμετρο	43
Εικόνα 25. Block και κώδικας παραδείγματος 3	44
Εικόνα 26. Έλεγχος σερβοκινητήρα με ποτενσιόμετρο	45
Εικόνα 27. Block και κώδικας παραδείγματος 4	46
Εικόνα 28. Εναλλακτική λύση παραδείγματος 4	47

Εισαγωγή

Η συνεχής ανάπτυξη της τεχνολογίας την σημερινή εποχή, συνιστά μια διαρκή προσπάθεια εξοικείωσης με αυτήν. Είτε για προσωπική χρήση, είτε για εκπαιδευτικό σκοπό, έχουν δημιουργηθεί διάφορα εργαλεία για να μπορέσει ο χρήστης να μάθει όσα χρειάζεται για την λειτουργία των ηλεκτρονικών συστημάτων που έχει στα χέρια του. Διαδραστικοί οδηγοί στο διαδίκτυο, σεμινάρια και αμέτρητα παραδείγματα , έχουν δημιουργηθεί με σκοπό την εξοικείωση μας στις νέες τάσεις της τεχνολογίας.

Το επίκεντρο της μελέτης μας, θα είναι ο προγραμματισμός του μικροελεγκτή Arduino στο περιβάλλον προγραμματισμού Scratch. Το Scratch δημιουργήθηκε για να αντικαταστήσει την πολυπλοκότητα των μεθόδων και των δομών σύνταξης της γλώσσας C++. Αυτό το πετυχαίνει αντικαθιστώντας το συντακτικό της γλώσσας με γραφικά σχέδια τα λεγόμενα blocks, τα οποία έχουν την μορφή εικονικής ψευδογλώσσας αλλά, εξυπηρετούν τις ίδιες ακριβώς αλγοριθμικές λειτουργίες με αυτήν.

Με μια σαφή οπτική ευκολία αλλά και κατανοητή παρουσίαση της γλώσσας προγραμματισμού, ο χρήστης είναι σε θέση να μάθει και να εξοικειωθεί με το Arduino με πιο ταχείς ρυθμούς, γλυτώνοντας έτσι την χρονοβόρα εκμάθηση της πολύπλοκης και δύσκολης για έναν αρχάριο, γλώσσας C++. Του προσφέρεται ένα μέσο πειραματισμού και άμεσης αλληλεπίδρασης, με τις πιο βασικές έως τις πιο πολύπλοκες λειτουργίες του Arduino.

Σκοπός αυτής της εργασίας, είναι να αναδείξουμε έναν ευχάριστο και εύκολο τρόπο διδασκαλίας Arduino. Η παρουσίαση βήμα-βήμα της διαδικασίας προγραμματισμού του και η παρατήρηση των αποτελεσμάτων, διευκολύνει τον αρχάριο μαθητή, να κατανοήσει τα θεμέλια της γλώσσας προγραμματισμού που θα κληθεί να χρησιμοποιήσει και ο ίδιος.

Ο αναγνώστης αρχικά έχει την ευκαιρία να μάθει κάποια εισαγωγικά στοιχεία για τα εργαλεία που καλείται να χρησιμοποιήσει. Το Arduino σαν σύστημα είναι έτοιμο για άμεση χρήση, αλλά το Raspberry pi διαθέτει μεγαλύτερη πολυπλοκότητα και συνοδεύεται από το δικό του λειτουργικό σύστημα. Για αυτό το λόγο θα δούμε βήμα-βήμα πως ετοιμάζουμε και ρυθμίζουμε κατάλληλα τον εξοπλισμό μας.

Εφόσον έχουμε τελειώσει με τις απαραίτητες προετοιμασίες και έχουμε δει τα γραφικά περιβάλλοντα του Arduino που λειτουργούν με τη μέθοδο Scratch, συμπεραίνουμε ότι, είτε ως τοπικές εφαρμογές είτε στο διαδίκτυο, η λειτουργία τους είναι ίδια και εξυπηρετούν τους ίδιους σκοπούς, μια πιο εύκολη και κατανοητή εναλλακτική προγραμματισμού του Arduino. Ένα από αυτά είναι τα περιβάλλοντα είναι το BlocklyDuino με το οποίο και θα ασχοληθούμε περισσότερο. Βήμα-βήμα από την εγκατάσταση μέχρι τα πρώτα αποτελέσματα με παραδείγματα, αποκτάμε μια πλήρη εικόνα των δυνατοτήτων του.

Η συγκεκριμένη πτυχιακή εργασία έχει την μορφή επιστημονικού οδηγού, έτσι ώστε ο αναγνώστης να μπορεί να αναπαράγει το περιεχόμενο της από μόνος του. Τέλος υπάρχουν διάφορα συμπεράσματα ως προς την λειτουργία και ανάπτυξη του πρότζεκτ, που πιθανώς να λύσουν σημαντικά ερωτήματα και απορίες.

1. Μικροελεγκτές-Μικροϋπολογιστές

1.1 Εισαγωγή στους μικροελεγκτές

Ο μικροελεγκτής είναι ένα είδος επεξεργαστή, πιο συγκεκριμένα ένας μικροεπεξεργαστής, ο οποίος μπορεί να λειτουργήσει με ελάχιστο αριθμό εξωτερικών εξαρτημάτων. [1] Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα (embedded) συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως για παράδειγμα σε συστήματα αυτοματισμού, οικιακές ηλεκτρικές συσκευές και κάθε είδους ηλεκτρικής συσκευής για ιδιωτική χρήση (ηλεκτρικά εργαλεία, ανιχνευτές μετάλλων κ.α.).

Η ανάπτυξη εφαρμογών με μικροελεγκτές καθίσταται άμεσα ως μια εναλλακτική λύση, η οποία μειώνει και απλοποιεί τα ψηφιακά κυκλώματα που απαιτούνται, καθώς το μεγαλύτερο μέρος αυτών αντικαθίσταται από εντολές κώδικα.

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα, δίνεται έμφαση στην επεξεργαστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά που διασυνδέονται με την κεντρική μονάδα, η οποία δεν είναι εξειδικευμένη.

Αντίθετα στους μικροεπεξεργαστές για ενσωματωμένα συστήματα, οι οποίοι έχουν μικρότερες δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, η ευελιξία είναι περιορισμένη, καθώς και η επεξεργαστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

1.1.1 Αναλυτικά τα πλεονεκτήματα των μικροελεγκτών είναι:

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν. Η ενσωμάτωση περιφερειακών συμβάλει επίσης στην μειωμένη κατανάλωση ισχύος από το σύστημα.
- Χαμηλότερο κόστος.
- Μεγαλύτερη αξιοπιστία λόγω των λιγότερων διασυνδέσεων.

- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από τον μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους – εξόδους.
- Μικρό μέγεθος συνολικού συστήματος.

Η επιτυχία μιας οικογένειας μικροελεγκτών καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές (compilers) από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή (assembly και γλώσσα μηχανής), προγραμματιστές της εσωτερικής μνήμης και εργαλεία αποσφαλμάτωσης (debuggers). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο από λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς.

Η πιο διαδεδομένη γλώσσα υψηλού επιπέδου για τον προγραμματισμό των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα και μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την assembly από τις περισσότερες εφαρμογές.

Από την άλλη μεριά, η γλώσσα assembly όπως και η γλώσσα μηχανής, είναι στενά συνδεδεμένες με την εσωτερική αρχιτεκτονική του μικροελεγκτή και βοηθούν τον τεχνικό να έχει πιο καθαρή εικόνα των χαρακτηριστικών και του σχεδιασμού του μικροελεγκτή και να αναπτύσσει πιο γρήγορες εφαρμογές. Επειδή όμως ο αριθμός των εντολών σε γλώσσα assembly είναι περιορισμένος κάνει αρκετά δύσκολο την ανάπτυξη σύνθετων εφαρμογών. Εδώ έρχονται πλέον οι γλώσσες υψηλού επιπέδου, όπως η C, όπου με την χρησιμοποίηση των βασικών δομών προγραμματισμού της ακολουθίας, της επιλογής και της επανάληψης βοηθούν τον τεχνικό στην ανάπτυξη εφαρμογών με περισσότερο κατανοητή δομή προγράμματος.

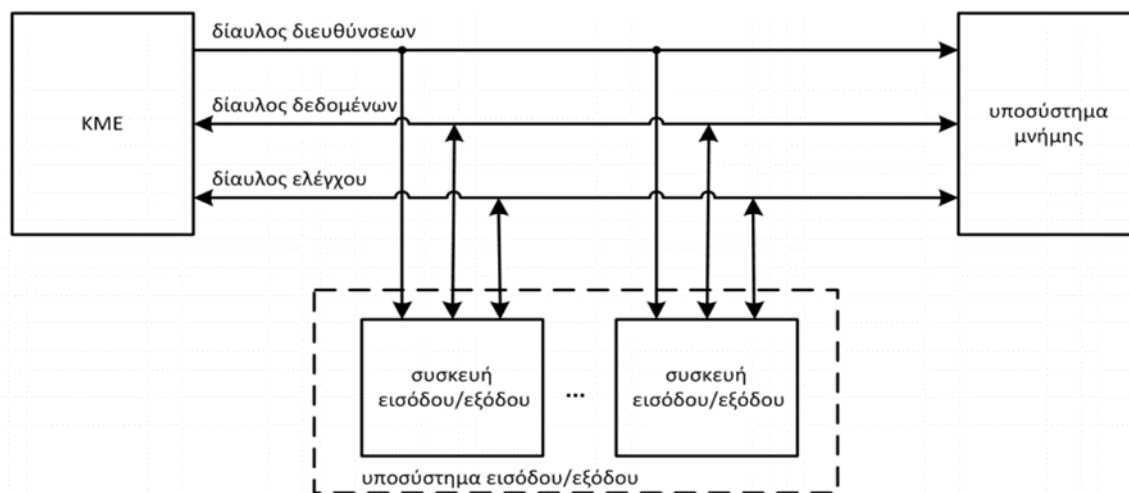
1.2 Η αρχιτεκτονική των μικροελεγκτών

Ένα τυπικό υπολογιστικό σύστημα αποτελείται από τις εξής βασικές μονάδες :

- Κεντρική Μονάδα Επεξεργασίας (Κ.Μ.Ε.)
- Μνήμη
- Κυκλώματα εισόδου/εξόδου (Input/Output – I/O)

Τα παραπάνω κυκλώματα συνδέονται μεταξύ τους και με όλα τα άλλα βοηθητικά, περιφερειακά κυκλώματα, μέσω των λεγόμενων διαδρόμων του υπολογιστή.

Στην περίπτωση που η Κ.Μ.Ε, υλοποιείται μέσω ενός μικροεπεξεργαστή, τότε το σύστημα θεωρείται ως «μικροϋπολογιστικό σύστημα», του οποίου η δομή φαίνεται στην εικόνα παρακάτω.



Εικόνα 1. Δομή μικροϋπολογιστικού συστήματος

1.2.1 Η Κεντρική Μονάδα Επεξεργασίας

Η Κεντρική Μονάδα Επεξεργασίας η οποία είτε είναι ενσωματωμένη στον μικροελεγκτή ή αποτελεί ξεχωριστό ολοκληρωμένο κύκλωμα, είναι η καρδιά του συστήματος και υλοποιείται με ένα ολοκληρωμένο κύκλωμα υπέρ – υψηλής κλίμακας ολοκλήρωσης (ULSI). Τα επιμέρους κυκλώματα που συγκροτούν εν γένει την κεντρική μονάδα επεξεργασίας είναι τα παρακάτω :

- Οι εσωτερικοί καταχωρητές
- Η αριθμητική και λογική μονάδα
- Η μονάδα ελέγχου

1.2.2 Η μνήμη

Μνήμη (memory) στα ψηφιακά ηλεκτρονικά είναι κάθε ηλεκτρονικό κύκλωμα που μπορεί να αποθηκεύσει ένα σύνολο από δυαδικά ψηφία (bits) τα οποία θα έχουν λογικές τιμές ‘0’ και ‘1’. Η ηλεκτρονική μνήμη RAM των σύγχρονων υπολογιστικών συστημάτων αποτελείται από διαδοχικά κύτταρα μνήμης σε διάταξη πίνακα οργανωμένα σε λέξεις πληροφορίας. Τα κύτταρα μνήμης είναι κατασκευασμένα με βάση τις αρχές της τεχνολογίας των ημιαγωγών. Σημαντικά χαρακτηριστικά που πρέπει να λαμβάνονται υπ’ όψη για τις μνήμες είναι :

- Η χωρητικότητα της μνήμης
- Ο χρόνος προσπέλασης

1.2.3 Οι Μονάδες Εισόδου/Εξόδου (I/O)

Με τους όρους μονάδες Εισόδου – Εξόδου (I/O) αναφερόμαστε σε όλες τις διατάξεις που επιτρέπουν τη μετατροπή όλων των μορφών πληροφορίας σε δυαδική μορφή και το αντίστροφο. Οι μονάδες I/O συνδέονται με το υπολογιστικό σύστημα μέσω των θυρών εισόδου/εξόδου, που μπορούν να θεωρηθούν ως επιπλέον καταχωρητές του συστήματος. Ο μικροελεγκτής ουσιαστικά βλέπει τις μονάδες I/O ως καταχωρητές στους οποίους γράφει δεδομένα, όταν πρόκειται για έξοδο ή διαβάζει δεδομένα όταν πρόκειται για είσοδο. Η επεξεργασία των δεδομένων, από την Κ.Μ.Ε., γίνεται σε μια σειρά από διακριτά βήματα, καθένα από τα οποία λέγεται ‘εντολή προγράμματος’. Επιπλέον, στην περίπτωση που το μικροϋπολογιστικό σύστημα είναι ανεπτυγμένο εξ’ ολοκλήρου μέσα σε ένα ολοκληρωμένο κύκλωμα, τότε αυτό το κύκλωμα λέγεται ‘μικροελεγκτής’.

Η βασική διαφορά ενός μικροελεγκτή από ένα μικροεπεξεργαστή είναι ότι ο πρώτος είναι ένα ολοκληρωμένο κύκλωμα (Integrated Circuit, IC) Ειδικού Σκοπού ενώ ο δεύτερος ένα ολοκληρωμένο κύκλωμα Γενικού Σκοπού. Συνήθως ο μικροελεγκτής

‘τρέχει’ ένα μόνο πρόγραμμα με το οποίο διαβάζει δεδομένα από συγκεκριμένες, τα επεξεργάζεται και τροφοδοτεί με αποτελέσματα συγκεκριμένες εξόδους. Τρέχει δηλαδή μόνο μία εφαρμογή και για το λόγο αυτό αποτελεί μια πολύ φθηνή, εύκολη και γρήγορη λύση σε εφαρμογές όπως συστήματα συναγερμών, οικιακών συσκευών κ.λπ.

1.3 Διαφορές Μικροελεγκτή-Μικροεπεξεργαστή

- Ο μικροεπεξεργαστής αποτελείται μόνο από μια κεντρική μονάδα επεξεργασίας, ενώ ο μικροελεγκτής περιέχει έναν επεξεργαστή, μνήμη, I / O όλα ενσωματωμένα σε ένα τσιπ.
- Ο μικροεπεξεργαστής χρησιμοποιείται σε προσωπικούς υπολογιστές, ενώ ο μικροελεγκτής χρησιμοποιείται σε ενσωματωμένο σύστημα.
- Ο μικροεπεξεργαστής χρησιμοποιεί έναν εξωτερικό δίαυλο για διασύνδεση με RAM, ROM και άλλα περιφερειακά, από την άλλη πλευρά, ο μικροελεγκτής χρησιμοποιεί έναν εσωτερικό δίαυλο ελέγχου.
- Οι μικροεπεξεργαστές βασίζονται στο μοντέλο Von Neumann ενώ Οι μικροελεγκτές βασίζονται στην αρχιτεκτονική του Χάρβαρντ
- Ο μικροεπεξεργαστής είναι περίπλοκος και ακριβός, με μεγάλο αριθμό οδηγιών για επεξεργασία, ενώ ο μικροελεγκτής είναι φθηνός και απλός με λιγότερες οδηγίες για επεξεργασία.

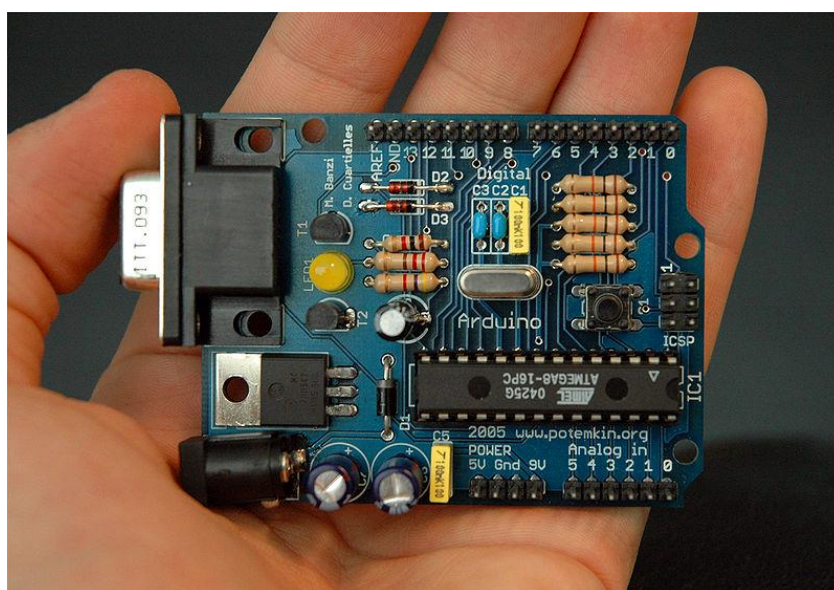
2. Το Arduino

2.1 Εισαγωγή στο Arduino



Το Arduino ανήκει στην κατηγορία των μικροελεγκτών μονής πλακέτας, πιο συγκεκριμένα είναι μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και συστήματα εισόδου/εξόδου, για άμεση επικοινωνία με τα αμέτρητα περιφερειακά του εξαρτήματα.

Το 2005 ξεκίνησε μια προσπάθεια να φτιαχτεί μια συσκευή ικανή για έλεγχο διαδραστικών προγραμμάτων από μαθητές, η οποία θα είναι πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την εποχή. Έτσι οι δημιουργοί του Arduino, Massimo Banzi και David Cueartielles αποφάσισαν να παράγουν μια νέα πλακέτα σε ένα μικρό εργοστάσιο στην Ιβρέα της Ιταλίας με το όνομα Αρντουίνο της Ιβρέας. Σύντομα είχαν δημιουργήσει κάτι τόσο επαναστατικό το οποίο άρχισε να εξελίσσεται με ταχύς ρυθμούς. [\[2\]](#)



Εικόνα 2. Μια πρόιμη πλακέτα Arduino

Σύγχρονα υπολογιστικά συστήματα όπως, μετεωρολογικοί σταθμοί, ανιχνευτές φωτιάς, καπνού και τοξικών αερίων, μπορούν να υλοποιηθούν χάρη στους ειδικά κατασκευασμένους αισθητήρες που διατίθενται στην αγορά. Εφαρμογές

αυτοματισμού, ανίχνευση κίνησης, έλεγχος της του τάσης ρεύματος, ανοίγουν τους ορίζοντες για μια πληθώρα εφαρμογών. Με προσπάθεια και λίγη φαντασία υπάρχει χώρος για νέες χρήσιμες ή διασκεδαστικές εφευρέσεις. Ανά τον κόσμο ερασιτέχνες και επαγγελματίες χρησιμοποιούν το Arduino καθημερινά στη ζωή τους, είτε για διασκέδαση, είτε για εργασία.

Μερικές πολύ γνωστές εφαρμογές είναι :

- Το [Arduboy](#) ,μια φορητή κονσόλα παιχνιδιών που βασίζεται στο Arduino.
- Το [ArduPilot](#) που αποτελεί λογισμικό και υλικό για drone.
- Το [C-STEM Studio](#), μια πλατφόρμα για πρακτική εκμάθηση υπολογιστών, τεχνολογίας, μηχανικής και μαθηματικών (C-STEM), με ρομποτική.
- Το [XOD](#), μια οπτική γλώσσα προγραμματισμού για το Arduino όπως το Scratch.

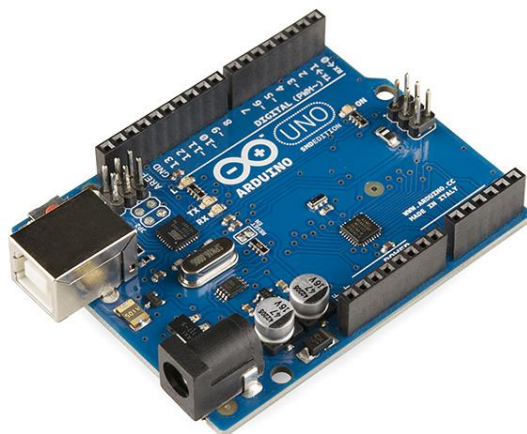
2.2 Τα είδη Arduino

Το Arduino πλέον έχει πάρει διάφορες μορφές, κάθε μία από αυτές έχει δημιουργηθεί για να αξιοποιεί πιο εύκολα τις διάφορες λειτουργίες του και να προσφέρει συμβατότητα και απόλυτη λειτουργικότητα στα περιφερειακά του. Είτε με περισσότερη επεξεργαστική ισχύ, είτε με μεγαλύτερο μέγεθος και περισσότερες εισόδους (pins), είναι σε θέση να εξυπηρετήσει κάθε πολύπλοκη εφαρμογή. [3]

Οι πιο βασικές και συνηθισμένες εκδόσεις Arduino είναι:

To Arduino UNO

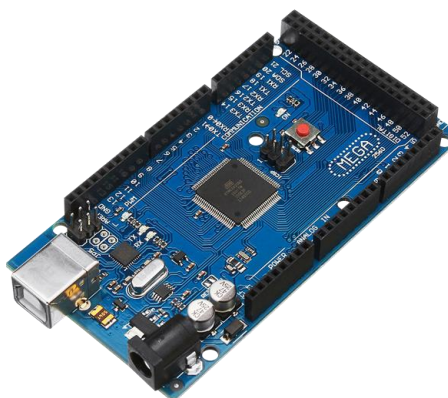
Είναι μια από τις καλύτερες εκδόσεις για αρχαίους. Εξαιρετικά φιλικό προς τον χρήστη και παράλληλα επιτρέπει εξαιρετική συνδεσιμότητα. Στην αγορά μπορούμε να βρούμε επίσης, έναν μεγάλο αριθμό αισθητήρων, εύκολων στην χρήση, για πειραματισμό και εξοικείωση μαζί του. Το [UNO](#) είναι το πιο διαδεδομένο μοντέλο παγκοσμίως και τα τελευταία χρόνια κατέχει την πρώτη θέση στην προτίμηση των ακαδημαϊκών οργανισμών.



Εικόνα 3. Το Arduino UNO

To Arduino Mega

Με έναν μεγάλο αριθμό εισόδων / εξόδων, το [Arduino Mega](#) είναι η ιδανική επιλογή για τους προχωρημένους λάτρεις του Arduino. Το συγκεκριμένο μοντέλο διαθέτει 16 αναλογικές ακίδες και μπορεί να επεκταθεί μέχρι στις 50 ή και παραπάνω. Διαθέτει περισσότερη μνήμη RAM και ROM από τις προηγούμενες πλακέτες της οικογένειας του Arduino. Είναι εξ ολοκλήρου μια πολύ καλή επιλογή για προχωρημένα πρότζεκτ όπως οι ρομποτικές κατασκευές.

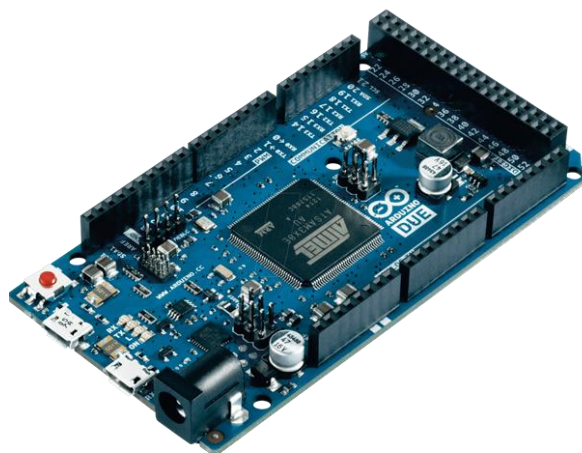


Εικόνα 4. Το Arduino Mega

Η μεγάλη επιτυχία και ζήτηση του Arduino, κατάφερε να το μετατρέψει σε ένα αναγκαίο εργαλείο. Πλέον είναι διαθέσιμα νέα πιο εξελιγμένα μοντέλα, με προ εγκατεστημένες λειτουργίες σύνδεσης στο internet με Wi-Fi ή μέσω θύρας ethernet, με ενσωματωμένη συσκευή ανάγνωσης microSD, με σειριακό δίαυλο που επιτρέπει την σύνδεση ψηφιακών συσκευών ήχου.

To Arduino Due

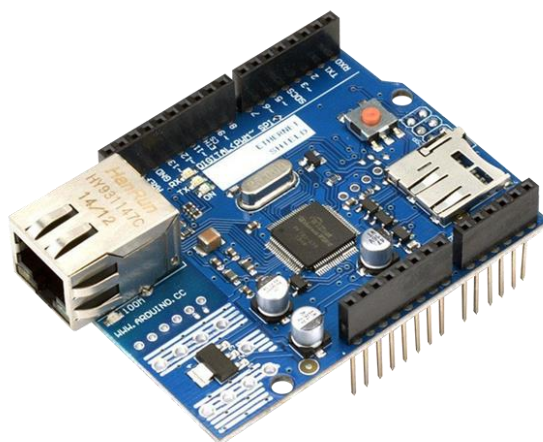
Πλέον αποτελεί το αποκορύφωμα της ισχύος και των δυνατοτήτων που παρέχονται από ένα Arduino. Με την μεγαλύτερη επεξεργαστική δύναμη και μνήμη προσφέρει μια εξαιρετική εμπειρία στους επαγγελματίες χρήστες. Διαθέτει 512KB ROM και 96KB RAM έτσι μπορεί να αντιμετωπίσει μεγάλα προγράμματα με υψηλή απαίτηση σε μνήμη. Είναι φανερό πως και το [Arduino Due](#) καλύπτει τις ανάγκες για πιο προχωρημένα πρότζεκτ.



Εικόνα 5. Το Arduino Due

To Arduino Ethernet REV 3

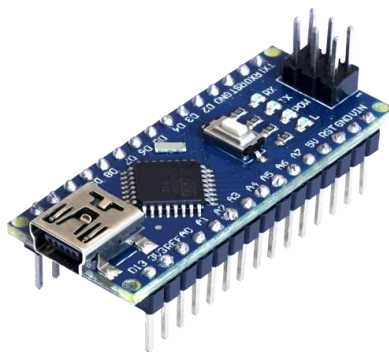
Αυτός ο μικροελεγκτής που βασίζεται στο μικροεπεξεργαστή [AT mega 328](#) διαθέτει 14 ψηφιακούς ακροδέκτες εισόδου – εξόδου, 6 αναλογικές εξόδους, έναν ταλαντωτή κρυστάλλου 16 MHz, υποδοχή τροφοδοσίας και μια υποδοχή RJ45 ή πιο κοινά θύρα Ethernet. Για πρότζεκτ Internet of Things, το συγκεκριμένο μοντέλο αποτελεί την ναυαρχίδα σε έλεγχο και δυνατότητες κατευθυνσιοδότησης διεργασιών και επικοινωνίας.



Εικόνα 6. Το Arduino Ethernet REV 3

To Arduino Nano

Το [Nano](#) αποτελεί την καταλληλότερη επιλογή για λειτουργία φορητών εφαρμογών. Σε αντίθεση με το UNO δεν διαθέτει υποδοχή τροφοδοσίας, αλλά μια θύρα USB και σε συνδυασμό με το μέγεθος του, είναι εξαιρετικό για φορητά πρότζεκτ. Η χαμηλή ισχύ που απαιτεί εξυπηρετεί εύκολα διάφορες εφαρμογές [IOT \(Internet of Things\)](#). Και αυτό επίσης αποτελεί μια πολύ καλή επιλογή για αρχαρίους.



Εικόνα 7. Το Arduino Nano

To Arduino MKR Zero

Εάν σας αρέσουν τα στοιχεία ήχου DIY , το [Arduino MKR Zero](#) είναι το καλύτερο για εσάς. Με την υποδοχή κάρτας SD για τη φιλοξενία αρχείων ήχου και τις διεπαφές SPI και I2S, το Arduino MKR Zero προσφέρει το τέλειο υλικό Arduino για εργασία με αρχεία ήχου. Ο ηλεκτρικός σειριακός διάυλος I2S είναι μια διεπαφή για τη σύνδεση ψηφιακών συσκευών και προσφέρει πολλές λειτουργίες. Υπάρχουν πολλές βιβλιοθήκες που εξυπηρετούν αυτό τον σκοπό.



Εικόνα 8. Το Arduino MKR Zero

Βλέπουμε λοιπόν πως έχουμε πολλές επιλογές και ακόμα περισσότερες. Είναι όλες διαθέσιμες στην αγορά αλλά και στην [επίσημη ιστοσελίδα](#) του Arduino. Όποιος και να είναι ο σκοπός και όσο πολύπλοκο να είναι το πρότζεκτ μας, πλέον τίποτα δεν είναι αδύνατο.

Ένα πολύ σημαντικό πλεονέκτημα με την συγκεκριμένη πλατφόρμα, είναι το γεγονός πως στο διαδίκτυο υπάρχουν αμέτρητα παραδείγματα και πολλές οδηγίες λειτουργίας εφ' όλης της ύλης. Δεν είναι λάθος να ψάχνουμε την γνώση και στη συνέχεια να την κάνουμε δική μας, για αυτό ακόμα και η πιο αλλόκοτη ιδέα μπορεί να υλοποιηθεί με λίγη παραπάνω προσπάθεια και ουσιαστικό ψάξιμο

2.3 Το λογισμικό



Όπως κάθε υπολογιστικό σύστημα, έτσι και το Arduino διαθέτει μια μέθοδο επικοινωνίας με τον χρήστη, την δική του γλώσσα και περιβάλλον για να μπορέσει να προγραμματιστεί και στη συνέχεια να εκτελέσει τον κώδικα.

Το περιβάλλον αυτό ονομάζεται Arduino Integrated Development Environment η αλλιώς Arduino IDE. Είναι μια εφαρμογή συμβατή με κάθε λειτουργικό σύστημα, χρησιμοποιεί μια σύνθεση της C και C++ γλώσσας προγραμματισμού και έχει την δυνατότητα άμεσης μεταφόρτωσης του κώδικα στην πλακέτα Arduino. Περιέχει έναν μεγάλο αριθμό βιβλιοθηκών λογισμικού, γεγονός που καθιστά πολλές λειτουργίες των εισόδων / εξόδων του Arduino διαθέσιμες στα πολυάριθμα περιφερειακά του. [4]

Η δομή του προγραμματισμού στο περιβάλλον IDE, είναι απλή καθώς χρειάζονται δυο βασικές συναρτήσεις μόνο, για να δημιουργηθεί ένα πρόγραμμα κυκλικής εκτέλεσης:

- **-setup()** : Συνάρτηση που εκτελείται μια φορά στην αρχή του προγράμματος και σκοπό έχει την αρχικοποίηση των μεταβλητών και διάφορων ρυθμίσεων.
- **-Loop()** : Συνάρτηση η οποία καλείται συνέχεια από το σύστημα μέχρι να απενεργοποιηθεί η πλακέτα. Περιέχει τον κορμό του προγράμματος εκεί όπου γράφεται η λειτουργία του.

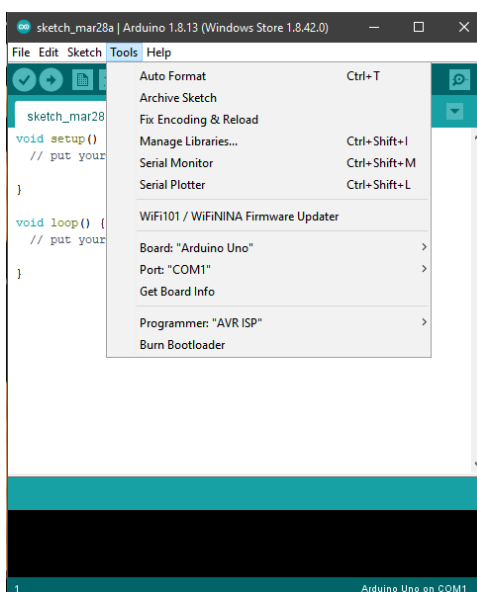
Ένα πολύ απλό παράδειγμα κώδικα είναι το παρακάτω και έχει την λειτουργία να αναβοσβήνει το ενσωματωμένο λαμπάκι LED στην πλακέτα με βήμα ενός δευτερολέπτου.

```
#define LED_PIN 13

void setup () {
  pinMode (LED_PIN, OUTPUT); // Ενεργοποίηση pin ως ψηφιακή έξοδο
}

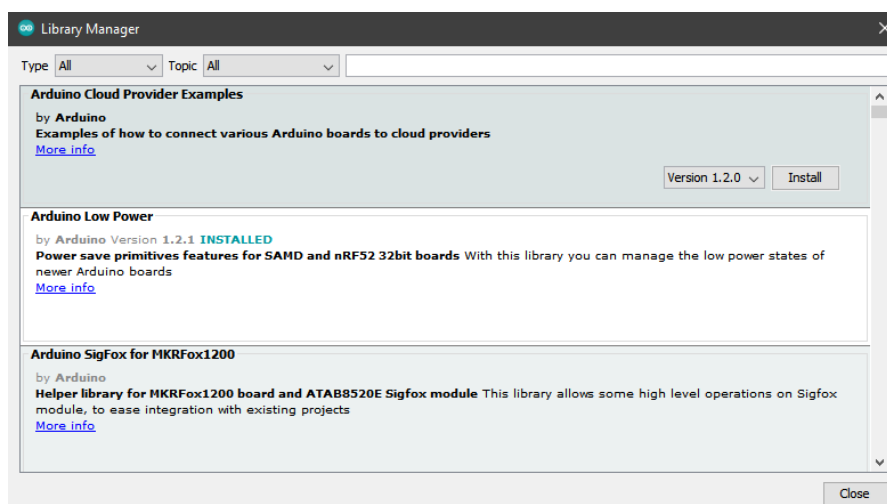
void loop () {
  digitalWrite (LED_PIN, HIGH); // Άναμμα LED
  delay (1000); // Αναμονή ενός δευτερολέπτου (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // Σβήσιμο LED
  delay (1000); // Αναμονή ενός δευτερολέπτου
}
```

Σε κάθε περίπτωση για να χρησιμοποιήσουμε το Arduino IDE, χρειάζεται να κατεβάσουμε και να συμπεριλάβουμε στο λογισμικό τις απαραίτητες βιβλιοθήκες που απαιτεί το εκάστοτε πρόγραμμα που θέλουμε να δημιουργήσουμε. Αυτή είναι μια πολύ εύκολη διαδικασία καθώς το IDE διαθέτει μια βάση δεδομένων (library manager) στην οποία μπορούμε να κάνουμε αναζήτηση των βιβλιοθηκών που χρειάζονται και έπειτα με ένα πάτημα να τις προσθέσουμε. Παρακάτω θα δούμε μια γρήγορη απεικόνιση αυτής της διαδικασίας. Αρχικά πηγαίνουμε στην καρτέλα Tools



Εικόνα 9. Το Arduino IDE

Και έπειτα επιλέγουμε Manage Libraries



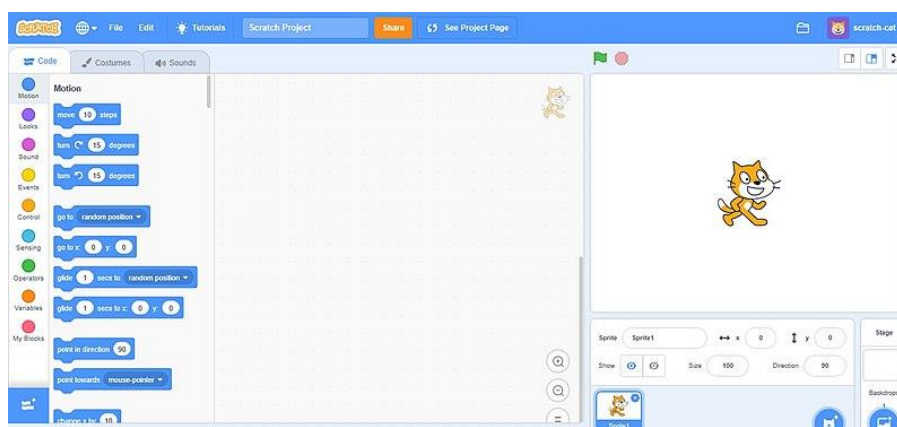
Εικόνα 10. Οι βιβλιοθήκες του IDE

2.4 To Scratch



Το [Scratch](#) είναι μια γλώσσα οπτικού προγραμματισμού βασισμένη στη χρήση μπλοκ γραφικών στοιχείων, που απευθύνεται σε μαθητές και ερασιτέχνες οποιασδήποτε ηλικίας, ως εκπαιδευτικό εργαλείο κωδικοποίησης. [5]

Η εταιρία MIT Media Lab άρχισε να αναπτύσσει την πλατφόρμα αυτή το 2003 αλλά το 2007 την έκανε διαθέσιμη στο κοινό. Πλέον το scratch έχει μεταφραστεί σε περισσότερες από 70 γλώσσες και χρησιμοποιείται σχεδόν σε όλο τον κόσμο. Διδάσκεται και χρησιμοποιείται σε κέντρα, σχολεία και κολέγια καθώς και σε δημόσια εκπαιδευτικά ιδρύματα με πάνω από 100 εκατομμύρια χρήστες τον χρόνο παγκοσμίως.



Εικόνα 11. Το Scratch

Η διεπαφή (interface) του Scratch έχει σχεδιαστεί με φιλικό τρόπο προς τον χρήστη και είναι χωρισμένο σε τρεις κύριες ενότητες.

- Την παλέτα των μπλοκ από όπου ο χρήστης με drag-n-drop προσθέτει στο σχέδιο του δίφορα στοιχεία.
- Την περιοχή κωδικοποίησης, το μέρος συναρμολόγησης και επεξεργασία των μπλοκ όπου σχηματίζεται το πρόγραμμα.
- Την περιοχή της σκηνής. Εμφανίζει τα αποτελέσματα του προγράμματος (π.χ. κινούμενα σχέδια, διάφορα γραφικά σχέδια).

3. Το Scratch για Arduino

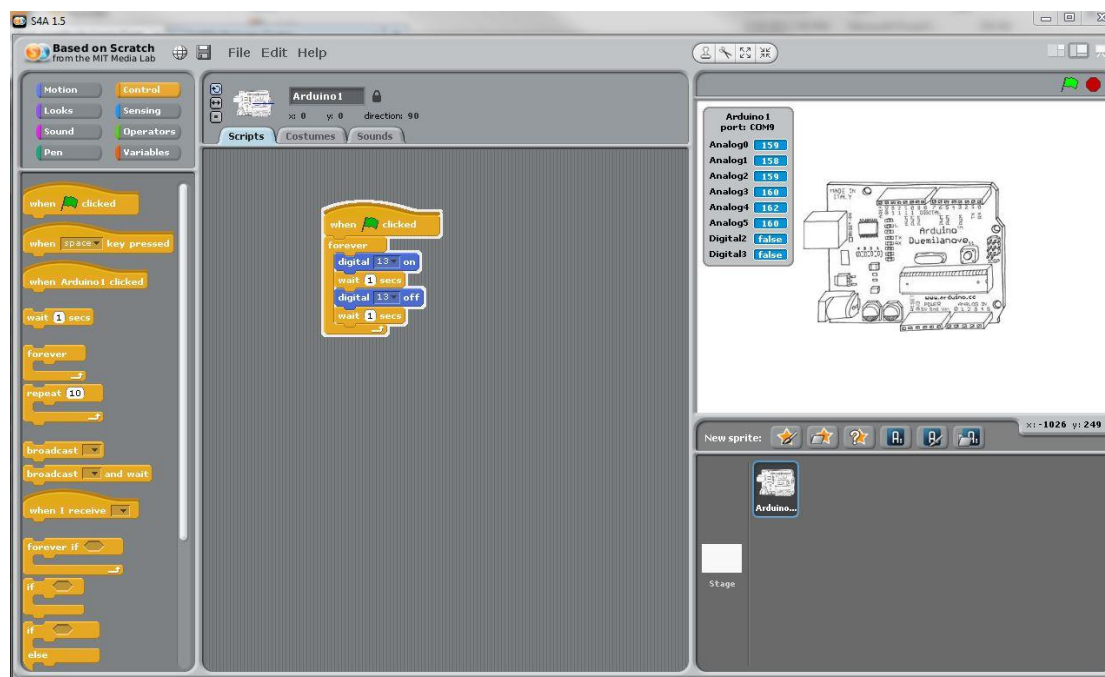
Το Scratch μπορεί να είναι δικαιωματικά ο πρωτοπόρος στον τομέα του οπτικού προγραμματισμού με μπλοκ, αλλά πλέον έχουν δημιουργηθεί πολλές παραλλαγές του για να ταιριάζει καλύτερα στον προγραμματισμό του Arduino, ή για να εξυπηρετεί οποιονδήποτε άλλο σκοπό. Αποτελεί έτσι και αλλιώς ένα εργαλείο διδασκαλίας ενός γενικευμένου συνόλου προγραμμάτων, όπως κινούμενα σχέδια, επεξεργασία αρχείων ήχου κ.α. για αυτό παρουσιάστηκε η ανάγκη για κάτι καινούργιο και πιο συμβατό με την πλακέτα.

3.1 Το S4A



Έτσι λοιπόν για να μετατραπεί το Scratch σε ένα περιβάλλον κατάλληλο για προγραμματισμό του Arduino, δημιουργήθηκε το [S4A](#). Μια τροποποίηση του πρώτου που επιτρέπει τον άμεσο προγραμματισμό της πλακέτας. Περιέχει νέα μπλοκ εντολών για τη διαχείριση αισθητήρων και άλλων διατάξεων που συνδέονται στο Arduino όπως μοτέρ, servo, κλπ. Στο περιβάλλον S4A, μια πλακέτα Arduino παριστάνεται από μία ειδική μορφή (sprite). Η μορφή αυτή αναζητά αυτόματα την θύρα USB που έχει συνδεθεί το Arduino. Είναι επίσης δυνατό να συνδεθούν πολλά Arduino ταυτόχρονα και τα οποία θα εμφανίζονται ως πρόσθετες διαφορετικές μορφές (sprites). [\[6\]](#)

Μπορούμε επίσης να χειριστούμε ασύρματα την πλακέτα τοποθετώντας ένα RF module σ' αυτή όπως το [Xbee](#). Το S4A επιτρέπει τον έλεγχο τόσων πλακετών Arduino όσων και οι θύρες USB που διαθέτουμε και είναι συμβατό με το Scratch, οπότε μπορούμε να ανοίγουμε πρότζεκτ του Scratch σ' αυτό. Επίσης η συμβατότητα δεν ισχύει και αντίστροφα, δηλαδή δεν μπορούμε στο Scratch να ανοίγουμε πρότζεκτ που φτιάξαμε στο S4A. Το πρωτόκολλο S4A αλληλοεπιδρά με το Arduino μεταδίδοντας την κατάσταση των διατάξεων και λαμβάνοντας την κατάσταση των αισθητήρων κάθε 75 ms, και για αυτό το εύρος των παλμών θα πρέπει να είναι μεγαλύτερο από αυτό το χρονικό διάστημα. Η ανταλλαγή των δεδομένων ακολουθεί το πρωτόκολλο του [PicoBoard](#) και απαιτεί ένα ειδικό πρόγραμμα (firmware) να εγκατασταθεί στην πλακέτα του Arduino.



Εικόνα 12. Το S4A

Το S4A είναι ένα πρόγραμμα διαθέσιμο για εγκατάσταση στον ηλεκτρονικό μας υπολογιστή και δυστυχώς δεν μπορεί να χρησιμοποιηθεί μέσω του δικτύου ή σε έναν τοπικό server, κάτι που απαιτείται στην συγκεκριμένη περίπτωση. Συμπερασματικά καταλήγουμε στα πλεονεκτήματα του τα οποία είναι:

- Εύκολο στη χρήση και γρήγορο στη λειτουργία του.
- Αυτόματη επαφή με το Arduino.
- Μεγάλος αριθμός στοιχείων μπλοκ.
- Δυνατότητα δημιουργίας δικών μας μπλοκ, αλλά και συμβολή στην ανάπτυξη νέων.
- Εκτέλεση σχεδίων Scratch.

Όμως από την άλλη πλευρά:

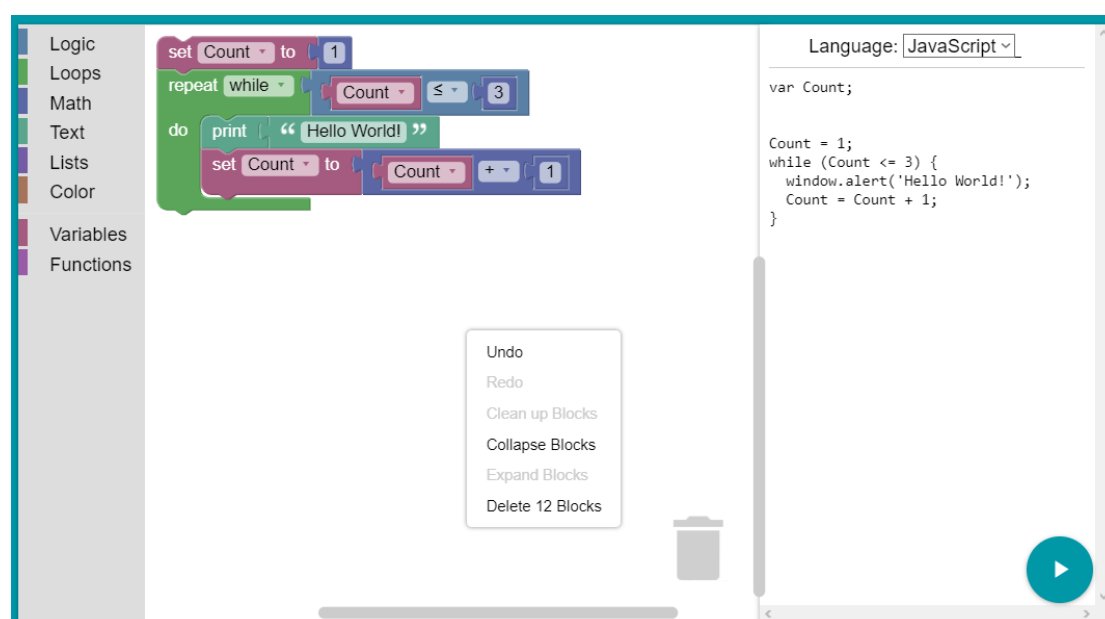
- Δεν μπορεί να λειτουργήσει μέσω τοπικού server.

3.2 Το Blockly

Το [Blockly](#) είναι ένα πρότζεκτ της εταιρίας Google, εξυπηρετεί εξ ολοκλήρου την σχεδίαση κώδικα για Arduino και είναι εμφανώς πιο απλό στη χρήση του. Επίσης είναι ανοιχτού κώδικα και μπορεί να χρησιμοποιηθεί μέσω browser. Και αυτό αποτελείται από μπλοκ γραφικά διευκολύνοντας έτσι τη δημιουργία κώδικα σε γλώσσες όπως JavaScript, Lua, Dart, PHP ακόμα και Python.

Η χρήση του είναι πιο εύκολη, καθώς ο χώρος εργασίας του αποτελείται από μια καρτέλα με τη λίστα των μπλοκ και τον χώρο συναρμολόγησης του κώδικα. Με drag and drop ο χρήστης μπορεί να αναδιατάξει εύκολα τα αντικείμενα της επιλογής του αλλά του παρέχεται και η δυνατότητα να δημιουργήσει νέα, δικά του μπλοκ.

Αυτό για να επιτευχθεί η Google έχει δημιουργήσει ένα εργαλείο που ονομάζεται Block Factory, το οποίο επιτρέπει την εύκολη δημιουργία νέων μπλοκ χρησιμοποιώντας τα ήδη υπάρχοντα. Βέβαια αυτό μπορεί να γίνει και με κώδικα JavaScript, αλλά το Block Factory κάνει απλούστερη την διαδικασία, του ορισμού των νέων μπλοκ αλλά και της δημιουργίας του κώδικα που εξυπηρετούν.



Εικόνα 13. Το Blockly


Πλέον υπάρχουν αμέτρητες παραλλαγές του, με την κάθε μια να εξειδικεύεται στις διάφορες προγραμματιστικές ανάγκες της σημερινής εποχής όπως είναι ο προγραμματισμός εφαρμογών για κινητά, τα παιχνίδια, γεννήτριες κώδικα ή ακόμα και δημιουργία νέων γλωσσών.

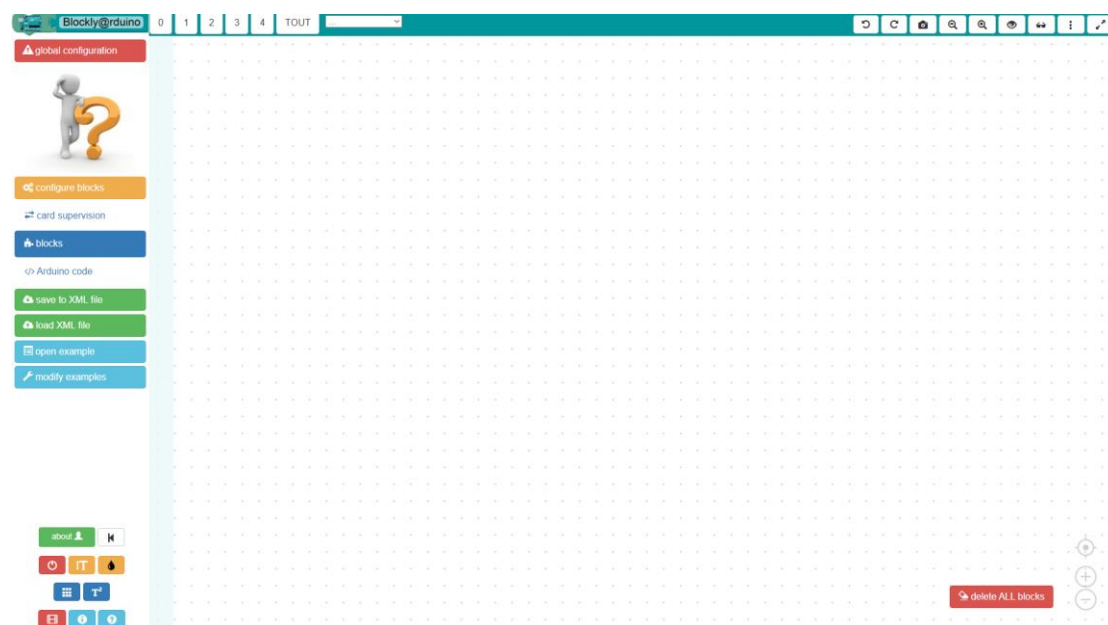
Στα πλεονεκτήματα του λοιπόν βρίσκουμε:

- Την ευελιξία του όσον αφορά τις προγραμματιστικές γλώσσες με τις οποίες συνδυάζεται.
- Το γεγονός πως λειτουργεί σε web browser.
- Την ευκολία στη χρήση του.
- Την δυνατότητα δημιουργίας νέων μπλοκ.

Για ακόμα μια φορά όμως, δεν μπορούμε να δουλέψουμε άμεσα με το Arduino και να ανεβάσουμε κώδικα σε αυτό μέσω web browser.

3.3 Το Blockly@rduino

 Για αυτό λοιπόν, έχουμε στην διάθεση μας ένα νέο και εξατομικευμένο εργαλείο για δημιουργία κώδικα Arduino με νέα εμφάνιση, νέες δυνατότητες και μεγάλη ισχύ.



Εικόνα 14. Το Blockly@rduino

Το [Blockly@rduino](#) έχει δημιουργηθεί με σκοπό να καταστήσει δυνατή την δημιουργία κώδικα με μπλοκ, μέσω browser και στη συνέχεια την αποστολή αυτού στο Arduino άμεσα και εύκολα. Οι επιλογές που μας παρέχει ξεπερνούν κατά πολύ σε ποικιλία, οποιοδήποτε άλλο περιβάλλον έχει φτιαχτεί για προγραμματισμό Arduino με Scratch.

Σαφώς είναι ένα πολύ δυνατό λογισμικό, όμως οι δημιουργοί του, επέλεξαν να το εμπορευματοποιήσουν και έτσι πλέον ο χρήστης πρέπει να αγοράσει συνδρομή για τα plug ins του ώστε να μπορέσει να ξεκλειδώσει λειτουργίες όπως το upload του κώδικα και διάφορες ρυθμίσεις που βοηθούν το κάθε πρότζεκτ.

Για αυτούς τους λόγους μπορούμε να συμπεράνουμε ότι, το Blockly@rduino

- Είναι ικανό να εκτελέσει σχεδόν οποιοδήποτε πρότζεκτ μπορεί να του ανατεθεί, χάρη στις αμέτρητες επιλογές που μας παρέχει.
- Μπορεί να κάνει upload τον κώδικα μας στο Arduino και λειτουργεί μέσω browser.

Αλλά το γεγονός πως χρειαζόμαστε συνδρομή για να μπορέσουμε να ανεβάσουμε τον κώδικα, είναι αρκετό να βάλει σε σκέψεις ακόμα και τον πιο απαιτητικό χρήστη. Αξίζει όμως να μάθουμε περισσότερα για αυτό. Θα εμβαθύνουμε αργότερα στη λειτουργία του και τα πολυάριθμα εργαλεία που μας παρέχει, επίσης μέσα από παραδείγματα θα καταλάβουμε πόσο εντυπωσιακά ικανό είναι να εκτελέσει μέχρι και τα πιο δύσκολα πρότζεκτ.

4. Το BlocklyDuino

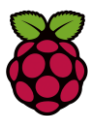
Το BlocklyDuino αποτελεί την καταλληλότερη επιλογή για την υλοποίηση του πρότζεκτ που παρουσιάζεται στην συγκεκριμένη πτυχιακή εργασία. Πρόκειται για μια παραλλαγή του Blockly από την Google. Όλα όσα χρειαζόμαστε από λειτουργίες (συνδεσιμότητα και upload κώδικα στο Arduino, επεξεργασία μέσω web browser), αλλά και ο αριθμός των block που μας παρέχει, καθιστά το στόχο μας εύκολο και λειτουργικό.



Εικόνα 15. Το BlocklyDuino

Δεν θα πούμε πολλά λόγια για το συγκεκριμένο λογισμικό καθώς θα ασχοληθούμε μαζί του στη συνέχεια. Θα δούμε πως εγκαθίσταται στο σύστημα μας, πως γίνεται η παραμετροποίηση του για συμβατότητα με τον εξοπλισμό μας, πως λειτουργεί, αλλά και παραδείγματα προγραμμάτων συνοδευόμενα από στιγμιότυπα κώδικα και καλωδίωσης.

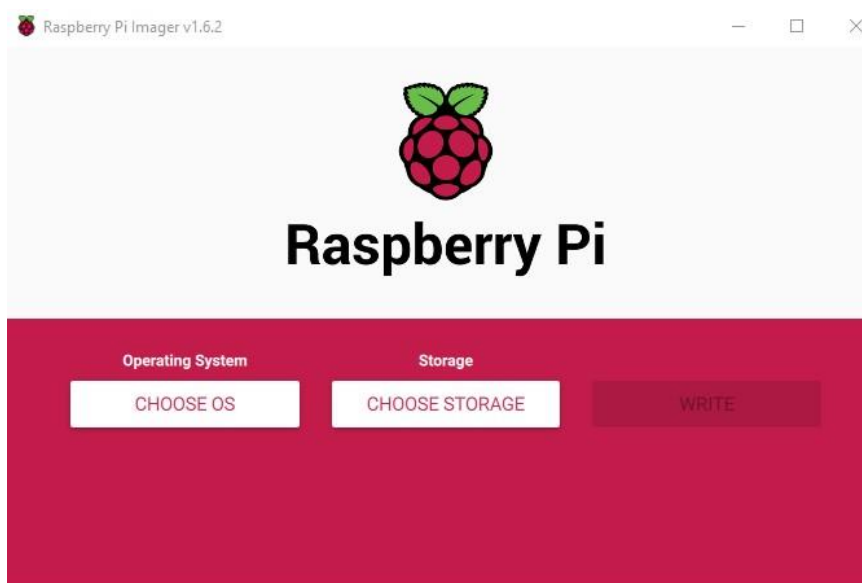
5. Το Raspberry Pi



Για να μπορέσουμε να ξεκινήσουμε το πρότζεκτ μας θα χρειαστεί να έχουμε στην κατοχή μας ένα raspberry pi με εγκατεστημένο λειτουργικό σύστημα. Κατά προτίμηση, το raspberry pi 4 με 4GB μνήμης είναι ικανό να καλύψει τις ανάγκες του project. Το λειτουργικό με το οποίο μπορούμε να εργαστούμε, μπορεί να είναι οποιοδήποτε αλλά στη συγκεκριμένη εργασία δουλεύουμε στο περιβάλλον [Raspbian](#).

Το Raspberry αποτελεί το βασικό πυλώνα για την υλοποίηση του πρότζεκτ μας. Πάνω σε αυτό θα λειτουργεί ο server μας και θα εκτελείται το πρόγραμμα BlocklyDuino. Αυτό που πρέπει να κάνουμε είναι να επισκεφτούμε την [επίσημη ιστοσελίδα](#) του και να κατεβάσουμε στον υπολογιστή μας το Raspberry Pi imager, ένα εργαλείο που διευκολύνει την εγκατάσταση του επιλεγμένου λειτουργικού συστήματος στην κάρτα micro SD, για τον σκοπό αυτό θα χρειαστούμε έναν micro-sd usb-adapter, που μπορούμε να βρούμε εύκολα στην αγορά.

Η λειτουργία του imager αποτελείται από δύο βήματα, αρχικά επιλέγουμε το λειτουργικό σύστημα που επιθυμούμε και μετά το αποθηκευτικό μέσο στο οποίο θέλουμε να εγκατασταθεί. Με το πέρας της εγκατάστασης, η κάρτα SD είναι έτοιμη για τοποθέτηση στη πλακέτα raspberry.



Εικόνα 16. Το Raspberry Pi imager

Μετά την πρώτη εκκίνηση του Raspberry Pi και εφόσον έχουμε πρόσβαση στο ίντερνετ, είναι απαραίτητο να εκτελέσουμε κάποιες εντολές για την αναβάθμιση του συστήματος μας. Αυτές είναι οι

```
sudo apt-get update
sudo apt-get upgrade
```

5.1 Λήψη και εγκατάσταση του Arduino IDE

Απαραίτητο είναι να κατεβάσουμε το λογισμικό [Arduino IDE](#), αν και μπορεί να μην το χρειαστούμε άμεσα, βοηθάει στη λειτουργία του BlocklyDuino, στο να δοκιμάσουμε νέους κώδικες και να κάνουμε απασφαλμάτωση ενός προγράμματος, πριν το υλοποιήσουμε μέσω BlocklyDuino. Για τον λόγο αυτό το κατεβάζουμε απευθείας από την επίσημη ιστοσελίδα του ή μέσω του terminal. για μεγαλύτερη ευκολία, με την εντολή

```
sudo apt-get install arduino
```

5.2 Λήψη και εγκατάσταση του BlocklyDuino

Το επόμενο βήμα εφόσον έχουμε έτοιμο το raspberry pi, είναι να κατεβάσουμε από το GitHub την Ελληνική έκδοση του BlocklyDuino από [εδώ](#) ή από την κονσόλα (terminal) του raspberry πληκτρολογώντας την εντολή:

```
sudo git clone https://github.com/kgioman/BlocklyDuino.git
```

Περιμένουμε να γίνει η λήψη και έπειτα κατευθυνόμαστε στον φάκελο pi του συστήματος μας όπου υπάρχει ο φάκελος BlocklyDuino. Για να έχουμε την δυνατότητα μετατροπής των περιεχομένων του φακέλου εκτελούμε από το terminal την εντολή `Chmod -R 777` (όνομα φακέλου) στη τοποθεσία όπου βρίσκεται ο φάκελος για παράδειγμα

```
sudo chmod -R 777 /home/pi/BlocklyDuino
```

5.2.1 Επεξεργασία αρχείων

Όπως είπαμε και πιο πριν, το πρόγραμμα μας για να εκτελεστεί επιτυχώς, χρειάζεται να προσαρμοστεί στο εκάστοτε σύστημα όπου φιλοξενείται. Έτσι λοιπόν θα δούμε πως και τι πρέπει να αλλάξουμε για το σκοπό αυτό. Ο φάκελος BlocklyDuino περιέχει αρκετά αρχεία τα οποία δεν θα χρειαστεί να ασχοληθούμε μαζί τους. Επιγραμματικά τα αρχεία με τα οποία θα δουλέψουμε είναι τα εξής

- arduino_web_server.py
- blockly-helper.js

όπως επίσης και το αρχείο συστήματος

- hosts

Η διαδικασία είναι πολύ απλή και σύντομη, το μόνο που μένει για να ξεκινήσουμε είναι να μάθουμε την IP του raspberry μας στο τοπικό δίκτυο και αυτό γίνεται μέσω ενός terminal εκτελώντας την εντολή

- hostname -i

Το αρχείο arduino_web_server.py το εντοπίζουμε αμέσως μόλις ανοίξουμε τον φάκελο BlocklyDuino, αμέσως πατάμε διπλό κλικ πάνω του και κατευθυνόμαστε στην γραμμή κώδικα 150 όπου και αντικαθιστούμε την αναγραφόμενη IP με την δικιά μας. Η τελική μορφή θα πρέπει να είναι

```
149 if __name__ == '__main__':
150     print "Το BlocklyDuino είναι έτοιμο στη διεύθυνση http://YOUR IP:8080/"
151     server = BaseHTTPServer.HTTPServer(("", 8080), Handler)
152     server.pages = {}
153     server.serve_forever()
```

Εικόνα 17. Στιγμιότυπο από το αρχείο web_server.py

Κάνουμε αποθήκευση και κλείσιμο και κατευθυνόμαστε στο αρχείο blockly-helper.js το οποίο βρίσκεται στη διαδρομή BlocklyDuino/blockly/apps/blocklyduino και το

ανοίγουμε με Text Editor. Κάπου προς το τέλος του αρχείου θα βρούμε την function `uploadCode`, στην οποία θα πρέπει να βάλουμε την IP μας για να γίνει σωστά η ανακατεύθυνση του κώδικα. Άρα η τελική εικόνα του κώδικα θα είναι

```
function uploadCode(code, callback) {
  var target = document.getElementById('content_arduino');
  var spinner = new Spinner().spin(target);

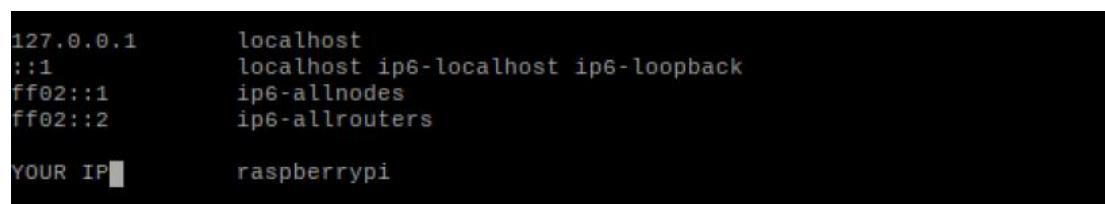
  var url = "http://YOUR IP:8080/";
  var method = "POST";
```

Εικόνα 18. Στιγμιότυπο από το αρχείο `blockly-helper.js`

Κάνουμε πάλι αποθήκευση και έξοδο πριν αλλάξουμε τα περιεχόμενα του τελευταίου μας αρχείου. Για ευκολία θα ανοίξουμε ένα terminal και θα πληκτρολογήσουμε την εντολή

```
sudo nano /etc/hosts
```

Στη τελευταία σειρά βάζουμε την IP μας δίπλα από το όνομα του `raspberrypi`, έτσι έχουμε



```
127.0.0.1      localhost
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
YOUR IP      raspberrypi
```

Εικόνα 19. Στιγμιότυπο από το αρχείο `hosts` μέσω terminal

Για να αποθηκεύσουμε πατάμε `control+X` και στη συνέχεια επιλέγουμε `YES`. Όλα είναι έτοιμα πλέον για να εκτελέσουμε το πρόγραμμα μας, μέσω terminal (το οποίο διατηρούμε ανοιχτό καθ' όλη τη διάρκεια), κατευθυνόμεστε στον φάκελο `BlocklyDuino` με `cd BlocklyDuino` και έπειτα εκτελούμε την εντολή

```
python arduino_web_server.py --port=/dev/ttyACM0
```

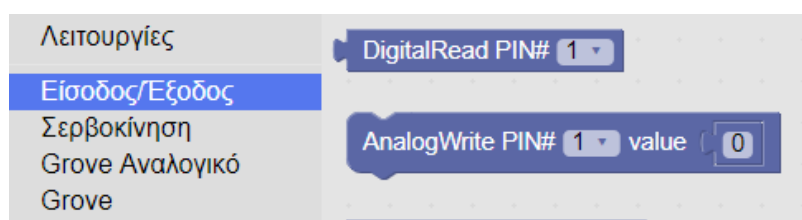
Νωρίτερα συναντήσαμε το `port 8080` στα αρχεία που επισκεφτήκαμε. Για τον λόγο αυτό λοιπόν, από τον browser της επιλογής μας επισκεπτόμαστε την σελίδα `YOUR IP:8080`.

6. Προγραμματισμός σε BlocklyDuino

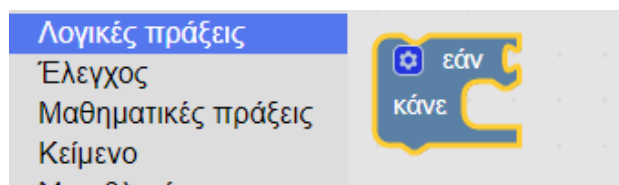
Το λογισμικό είναι έτοιμο πλέον για χρήση. Όπως είχαμε συμφωνήσει, θα δούμε την εφαρμογή μας πιο λεπτομερώς από εδώ και πέρα. Με παραδείγματα και εικόνες συνδεσμολογίας θα κατανοήσουμε τις βασικές αρχές του. Θα χρησιμοποιήσουμε το πρόγραμμα Fritzing, για να περιγράψουμε οπτικά τα κυκλώματά μας.

Παράδειγμα 1. Λειτουργία LED

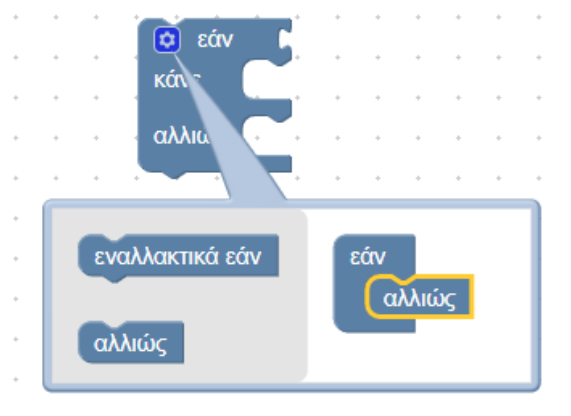
Ας ξεκινήσουμε λοιπόν με ένα πολύ απλό παράδειγμα, που η λειτουργία του θα είναι να ανάβει και να σβήνει ένα LED, κατά το πάτημα ενός κουμπιού. Το Arduino πρέπει να διαβάζει συνεχώς την κατάσταση στη θέση 9, και αυτό γίνεται μέσω του block DigitalRead.



Στη συνέχεια ελέγχουμε αν η τιμή που λαμβάνεται από το block αυτό είναι HIGH ή LOW. Εάν η τιμή είναι HIGH το LED ανάβει διαφορετικά σβήνει. Αυτός ο έλεγχος θα γίνεται με μια μέθοδο EAN με το αντίστοιχου block.



Στο block EAN από προεπιλογή δεν υπάρχει η συνθήκη ΑΛΛΙΩΣ για αυτό κάνουμε κλικ στην επιλογή Ρυθμίσεις και σέρνουμε την επιλογή μας στο αρχικό block.



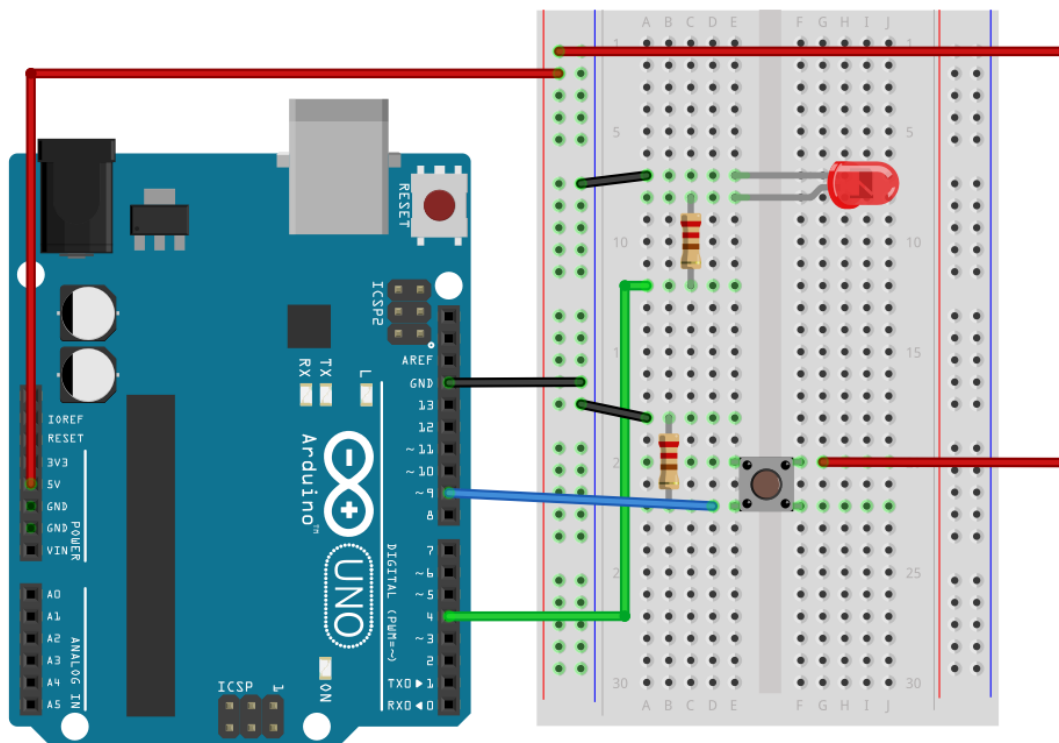
Ακολουθεί ένα block σύγκρισης, το οποίο επιστρέφει μια τιμή HIGH ή LOW ανάλογα με τις παραμέτρους που του θέτουμε. Στην προκειμένη περίπτωση θα συγκρίνουμε την τιμή του DigitalRead με μια είσοδο HIGH.



Τα υλικά που θα χρησιμοποιήσουμε είναι:

- Arduino UNO Rev3
- Breadboard 400 point
- Καλώδια
- LED
- Κουμπί απλής ώθησης
- Αντίσταση 10K (για το κουμπί), αντίσταση 220 Ohm (για το LED)

Πάμε να δούμε αρχικά την απεικόνιση του κυκλώματος με την βοήθεια του fritzing.



Εικόνα 20. Λειτουργία LED μέσω button

Στη συνέχεια δημιουργούμε τα γραφικά (block) μας στο BlocklyDuino. Κανονικά για να εκτελέσουμε το παραπάνω παράδειγμα, θα γράφαμε στο περιβάλλον του Arduino IDE τον παρακάτω κώδικα.

```
const int buttonPin = 9;
const int ledPin = 4;
void setup () {
  pinMode (ledPin, OUTPUT);
  pinMode (buttonPin, INPUT);
}
void loop () {
  if (digitalRead(buttonPin) == HIGH) {
    digitalWrite (ledPin, HIGH);
  } else {
    digitalWrite (ledPin, LOW);
  }
}
```

Αλλά όταν προγραμματίζουμε με BlocklyDuino, ο κώδικας παράγεται αυτόματα και για του λόγου το αληθές, το τελικό πρόγραμμα θα δείχνει όπως παρακάτω:

The image shows two screenshots of the BlocklyDuino web application. The top screenshot displays a visual block-based program on a grid. It features an 'if-else' block structure. The 'if' block is labeled 'εάν' (if) and contains a 'DigitalRead PIN# 9' block with a dropdown set to 'HIGH'. The 'then' block is labeled 'κάνε' (do) and contains a 'DigitalWrite PIN# 4 Stat HIGH' block. The 'else' block is labeled 'αλλιώς' (otherwise) and contains a 'DigitalWrite PIN# 4 Stat LOW' block. The bottom screenshot shows the same program converted into C++ code. The code is as follows:

```
void setup()
{
  pinMode(9, INPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  if (digitalRead(9) == HIGH) {
    digitalWrite(4, HIGH);
  } else {
    digitalWrite(4, LOW);
  }
}
```

Εικόνα 21. Block και κώδικας παραδείγματος 1

Εφόσον έχουμε υλοποιήσει στο Arduino μας το κύκλωμα μας σύμφωνα με το σχηματικό από το fritzing (βλ. **Εικόνα 20**), μένει να κάνουμε ανέβασμα του κώδικα στο Arduino πατώντας Ανέβασμα αρχείου. Αμέσως θα δούμε στο terminal του raspberry τη διαδικασία αυτή.

```

Uploading...
192.168.2.6 - - [07/Oct/2021 18:38:24] "POST / HTTP/1.1" 200 -
sketch to upload: void setup()
{
  pinMode(9, INPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  if (digitalRead(9) == HIGH) {
    digitalWrite(4, HIGH);
  } else {
    digitalWrite(4, LOW);
  }
}
}
created sketch at /tmp/tmpCDJA0k/tmpCDJA0k.ino
Uploading with arduino --upload --port /dev/ttyACM0 /tmp/tmpCDJA0k/tmpCDJA0k.ino
Picked up JAVA_TOOL_OPTIONS:
Loading configuration...
Initializing packages...

```

Μόλις εμφανιστεί στην οθόνη του υπολογιστή μας το μήνυμα ολοκλήρωσης φόρτωσης, το πρόγραμμα μας είναι πλέον λειτουργικό.

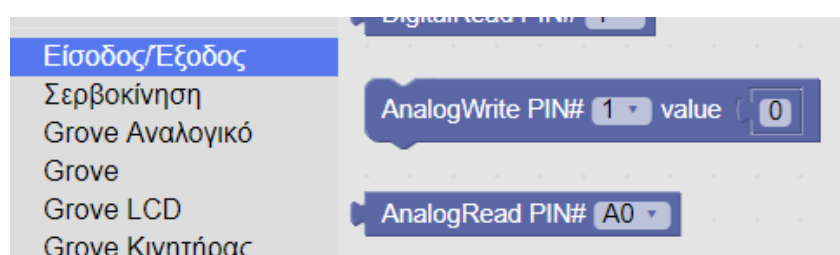
Είναι ολοφάνερο πως με ιδιαίτερη ευκολία υλοποιήσαμε το πρώτο ολοκληρωμένο πρόγραμμα στο περιβάλλον του BlocklyDuino, εστιάζοντας σε κάποιες από τις πιο βασικές συναρτήσεις της γλώσσας του προγραμματισμού Arduino.

ΠΡΟΣΟΧΗ!. Σε αυτό το σημείο αξίζει να αναφερθεί, το γεγονός πως μπορεί να έρθουμε αντιμέτωποι με διάφορα σφάλματα κατά το ανέβασμα του κώδικα. Η πιο πιθανή αιτία αυτού εάν και εφόσον έχουμε ακολουθήσει σωστά τα παραπάνω βήματα, είναι η αυθεντικότητα της πλακέτας Arduino που χρησιμοποιούμε. Στην αγορά υπάρχουν πολλές απομιμήσεις Arduino οι οποίες τις περισσότερες φορές ξεχωρίζουν στο μάτι από τις αυθεντικές. Σε [αυτή τη σελίδα](#) μπορούμε να μάθουμε πώς να τις διακρίνουμε.

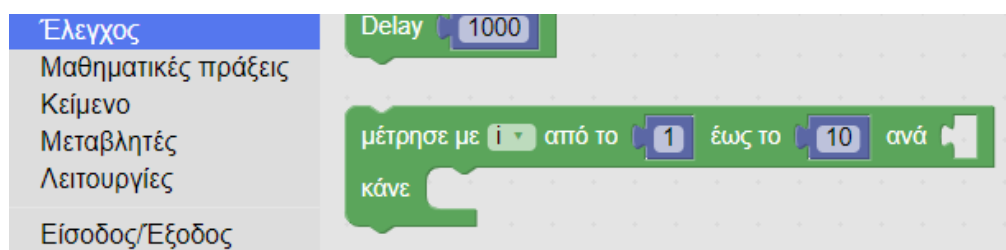
Παράδειγμα 2. Αλλαγή φωτεινότητας LED (fading)

Αυτή τη φορά θα δούμε πως μπορούμε να ελέγξουμε τις αναλογικές εξόδους ούτως ώστε να αλλάξουμε την φωτεινότητα ενός LED, με τιμές από τα χαμηλά προς τα υψηλά και το αντίθετο. Για να επιτευχθεί αυτό, πρέπει να αλλάξουμε την ένταση της λυχνίας LED σε κάθε βήμα με σταθερά βήματα.

Θα χρησιμοποιήσουμε το block AnalogWrite, το οποίο εξυπηρετεί την έννοια της διαμόρφωσης πλάτους παλμών (PWM). Ένα σήμα PWM είναι η προσομοίωση αναλογικής εξόδου με ψηφιακό τρόπο. Πιο συγκεκριμένα το σήμα αυτό μπορεί να έχει 255 στάθμες τάσης.



Επόμενο είναι το block βρόχου FOR (μέτρησε με), αυτό ζητά δύο ορίσματα το αρχικό και τα τελικό σημείο μιας μεταβλητής καταμέτρησης. Αρχίζει να αυξάνει ή να μειώνει τον μετρητή ανάμεσα στα δυο αυτά σημεία, δηλαδή της τάσης του ρεύματος και κατά συνέπεια της φωτεινότητας του LED.

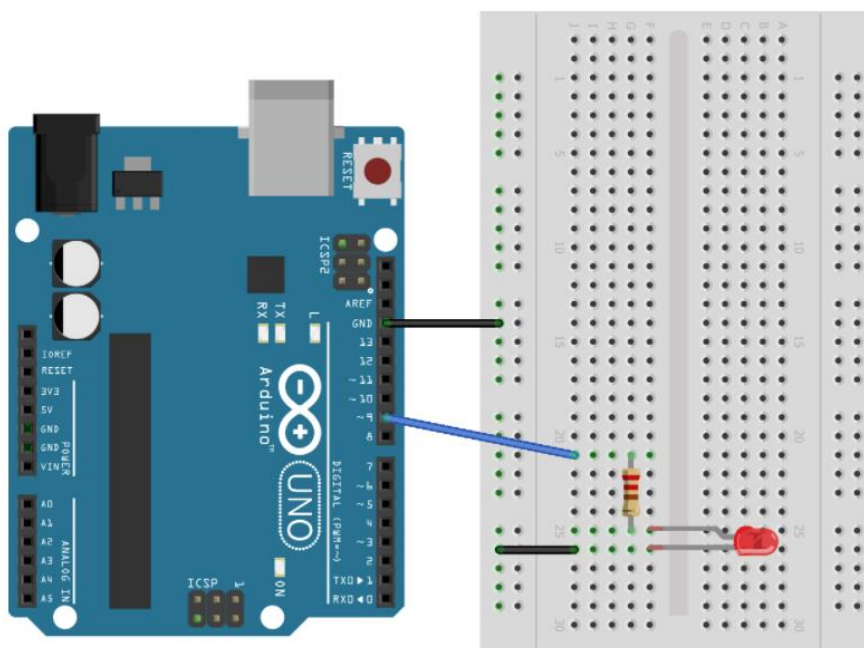


Θέτουμε το αρχικό σημείο σε 0 και το τελικό σε 255. Χρησιμοποιώντας την μεταβλητή *i* (μετρητής), αλλάζουμε την τιμή PWM του LED.

Τα υλικά που θα χρησιμοποιήσουμε είναι:

- Arduino UNO Rev3
- Breadboard 400 point
- Καλώδια
- LED
- Αντίσταση 220 Ohm

Η απεικόνιση του κυκλώματος στο fritzing.



Εικόνα 22. Αλλαγή φωτεινότητας LED

Η διαδικασία που ακολουθούμε είναι η ίδια, επιλέγουμε τα block που χρειάζονται και συνθέτουμε το πρόγραμμα μας.

Blockly Duino

Κελιά Arduino XML Ανέβασμα αρχείου

- Λογικές πράξεις
- Έλεγχος
- Μαθηματικές πράξεις
- Κείμενο
- Μεταβλητές
- Λειτουργίες
- Είσοδος/Εξοδος
- Σερβοκίνηση
- Grove Αναλογικό
- Grove
- Grove LCD
- Grove Κινητήρας

```

    μέτρησε με i από το 0 έως το 255 ανά
    κάνε
      AnalogWrite PIN# 9 value i
      Delay 30
    μέτρησε με i από το 255 έως το 0 ανά
    κάνε
      AnalogWrite PIN# 9 value i
      Delay 30
  
```

Και έπειτα παρατηρούμε πάλι στην καρτέλα Arduino τον αυτοματοποιημένο κώδικα σε πιο απλοποιημένη μορφή από ότι έχουμε συνηθίσει γράφοντας τον στο χέρι.



```
int i;

void setup()
{

}

void loop()
{
  for (i = 0; i <= 255; i++) {
    analogWrite(9, i);
    delay(30);
  }
  for (i = 255; i >= 0; i--) {
    analogWrite(9, i);
    delay(30);
  }
}
```

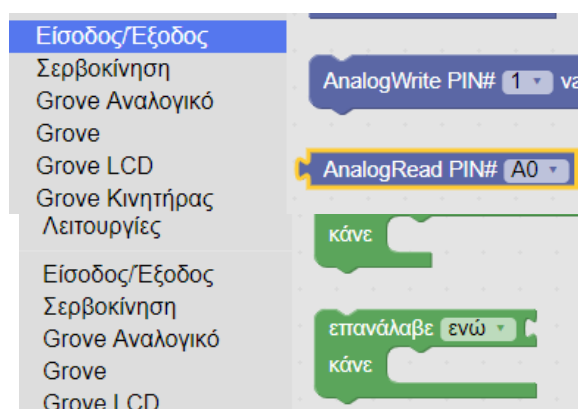
Εικόνα 23. Block και κώδικας παραδείγματος 2

Παράδειγμα 3. Λειτουργία LED με ποτενσιόμετρο ως διακόπτης

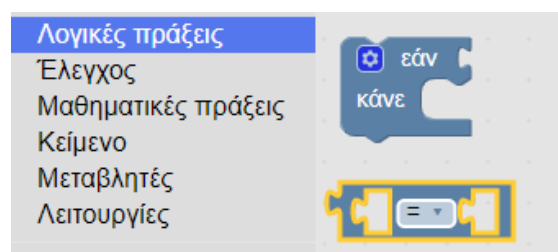
Σε αυτή την άσκηση θέλουμε να ελέγχουμε χειροκίνητα τη φωτεινότητα του LED, πώς μπορεί να επιτευχθεί αυτό;. Με την προσθήκη ενός ποτενσιόμετρου στο κύκλωμα μας.

Το ποτενσιόμετρο είναι μια μεταβλητή αντίσταση που χρησιμοποιείται σαν διαιρέτης τάσης. Έχει τρία τερματικά-ακροδέκτες, το μεσαίο τερματικό είναι ρυθμιστικό και τα άλλα δύο είναι για τροφοδοσία (Vcc & Gnd). Το ρυθμιστικό άκρο έχει την δυνατότητα να σπάει την συνολική αντίσταση του ποτενσιόμετρου ανάλογα με την γωνία θέσης στην οποία βρίσκεται, άρα να ελέγχει έτσι την διαφορά δυναμικού του κυκλώματος. Για τον λόγο αυτό παίρνει την ονομασία του διαιρέτη τάσης.

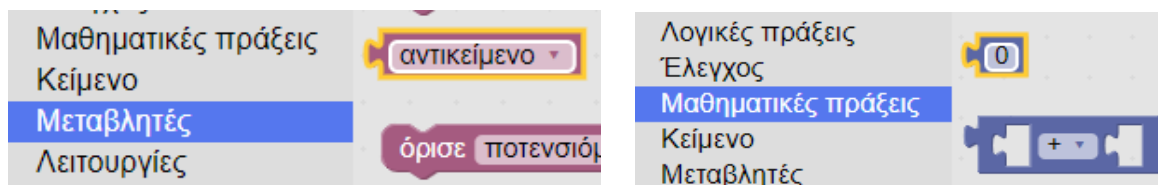
Το ADC του Arduino μπορεί να διαβάζει αυτή την τάση μέσω μιας αναλογικής εισόδου (A0,A1 κοκ) και επιστρέφει μια τιμή στην περιοχή 0-1023. Έτσι χρησιμοποιούμε το block AnalogRead και τον βρόχο WHILE (επανάλαβε) για να παρακολουθήσουμε αυτή την ένδειξη.



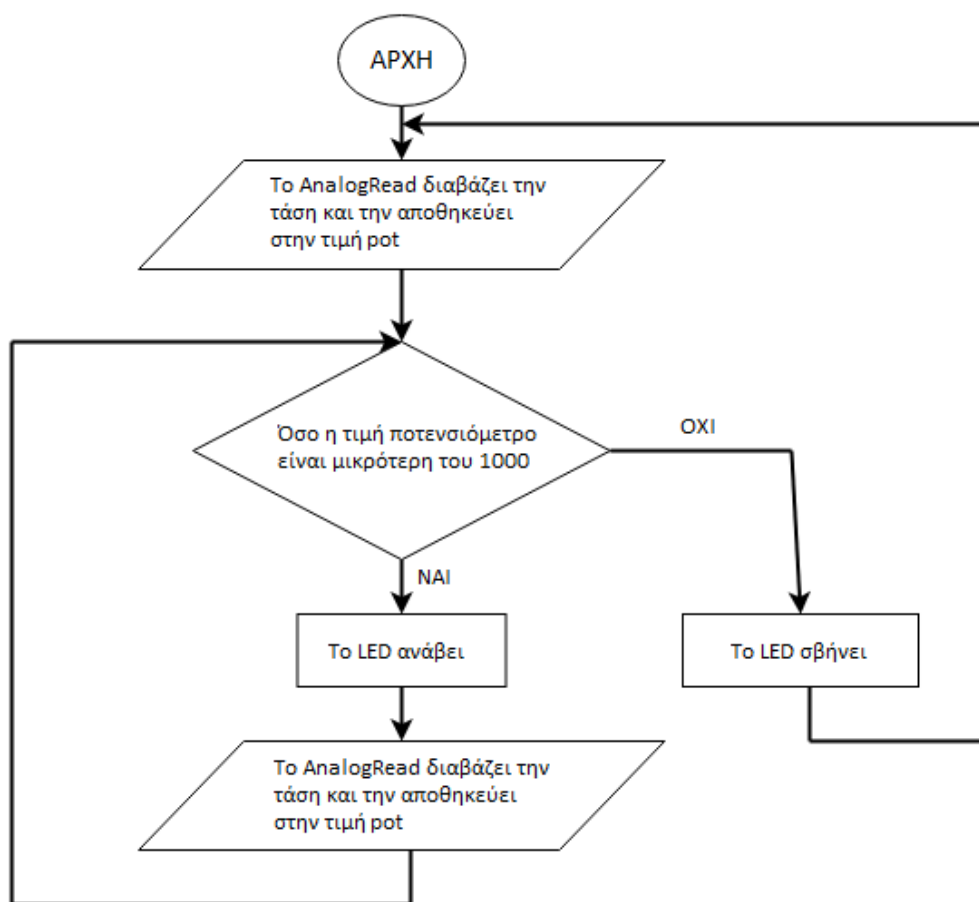
Πρέπει να καθορίσουμε τον σωστό αριθμό pin στη συνάρτηση AnalogRead και στον βρόχο WHILE ο οποίος θα επαναλαμβάνεται συνεχώς, να εισάγουμε ένα block συνθηκών από την καρτέλα Λογικές πράξεις.



Θα χρειαστούμε επίσης ένα block αντικειμένου και ένα block τιμής.



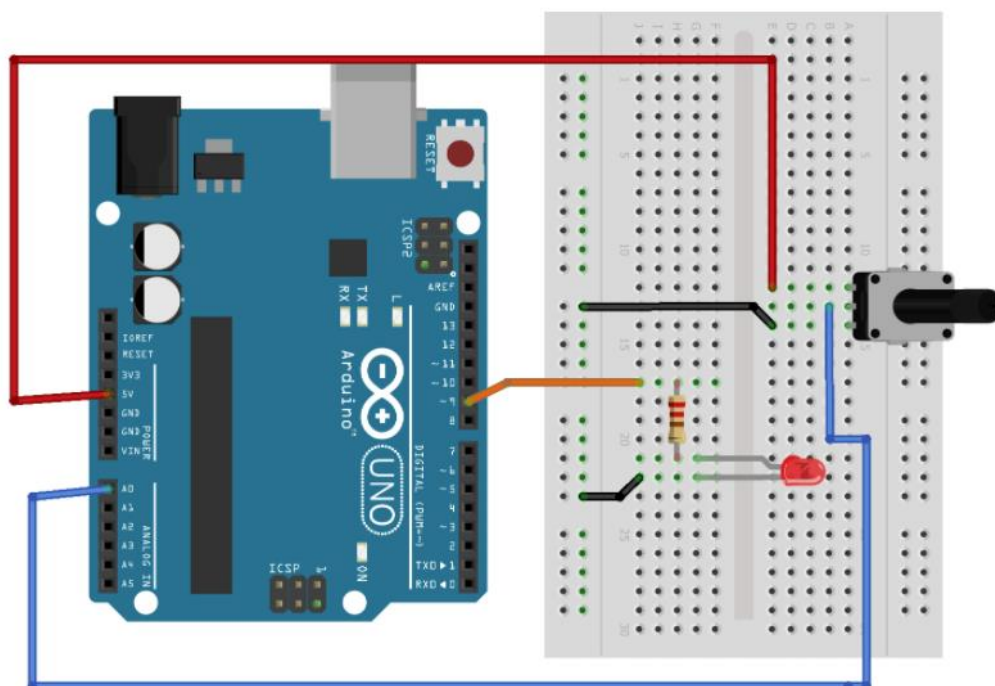
Για καλύτερη κατανόηση του τι θέλουμε να επιτύχουμε ας ρίξουμε μια ματιά στο παρακάτω διάγραμμα ροής.



Τα υλικά που θα χρησιμοποιήσουμε είναι:

- Arduino UNO Rev3
- Breadboard 400 point
- Καλώδια
- LED
- Αντίσταση 220 Ohm
- Ποτενσιόμετρο μικρής αντίστασης

Ας δούμε αρχικά μια απεικόνιση του κυκλώματος στο fritzing.



Εικόνα 24. Αλλαγή φωτεινότητας LED με ποτενσιόμετρο

Το τελικό πρόγραμμα μας θα δείχνει όπως παρακάτω. Εφόσον έχουμε ετοιμάσει το κύκλωμα μας πατάμε Ανέβασμα αρχείου και μετά από λίγο το παράδειγμα μας είναι λειτουργικό.

Blockly Duino

Κελιά	Arduino	XML	Ανέβασμα α
<ul style="list-style-type: none"> Λογικές πράξεις Έλεγχος Μαθηματικές πράξεις Κείμενο Μεταβλητές Λειτουργίες 	<pre> όρισε pot μέχρι το AnalogRead PIN# A0 επανάλαβε ενώ pot ≤ 1000 κάνε όρισε pot μέχρι το AnalogRead PIN# A0 DigitalWrite PIN# 9 Stat HIGH DigitalWrite PIN# 9 Stat LOW </pre>		

The screenshot shows the Blockly Duino interface with the 'Arduino' tab selected. The code editor contains the following C++ code:

```
int pot;

void setup()
{
  pinMode(9, OUTPUT);
}

void loop()
{
  pot = analogRead(A0);
  while (pot <= 1000) {
    pot = analogRead(A0);
    digitalWrite(9, HIGH);
  }
  digitalWrite(9, LOW);
}
```

Εικόνα 25. Block και κώδικας παραδείγματος 3

Σαν επόμενο βήμα από το παράδειγμα που μόλις είδαμε, θα μπορούσαμε να αλλάξουμε τον κώδικα μας ώστε να μπορεί το ποτενσιόμετρο να ελέγχει την φωτεινότητα του LED για ένα ελεγχόμενο fading.

The screenshot shows the Blockly Duino interface with the 'Arduino' tab selected. The workspace contains three blocks:

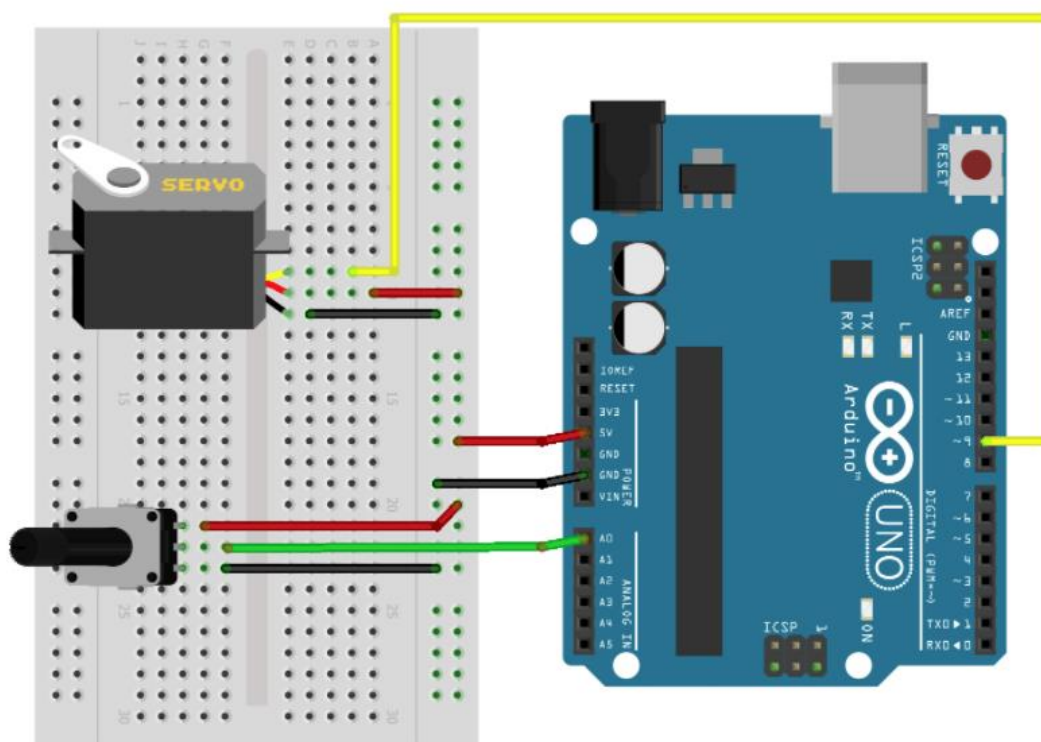
- An 'AnalogRead PIN#' block with 'A0' selected.
- A 'writeValue' block with 'readValue' selected and a division operator '÷' followed by the number '4'.
- An 'AnalogWrite PIN#' block with '9' selected and 'value' selected.

The left sidebar shows a list of categories: Λογικές πράξεις, Έλεγχος, Μαθηματικές πράξεις, Κείμενο, Μεταβλητές, and Λειτουργίες.

Παράδειγμα 4. Έλεγχος σερβοκινητήρα με ποτενσιόμετρο

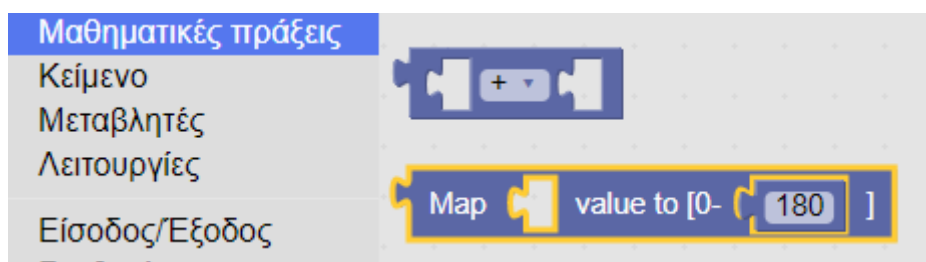
Στο προηγούμενο παράδειγμα συζητήσαμε για το ποτενσιόμετρο και πως διαβάζουμε την τιμή του χρησιμοποιώντας το block AnalogRead. Αυτή τη φορά θα χρησιμοποιήσουμε τις τιμές του (0-1023) για να ορίσουμε την θέση ενός σερβοκινητήρα.

Ο σερβοκινητήρας έχει τρία άκρα την είσοδο Vcc, την γείωση Gnd και το PWM. Επίσης στο πρόγραμμα μας θα παίρνει εύρος τιμών ανάμεσα στις γωνίες 0-180 μοίρες. Ας δούμε το διάγραμμα του κυκλώματος.

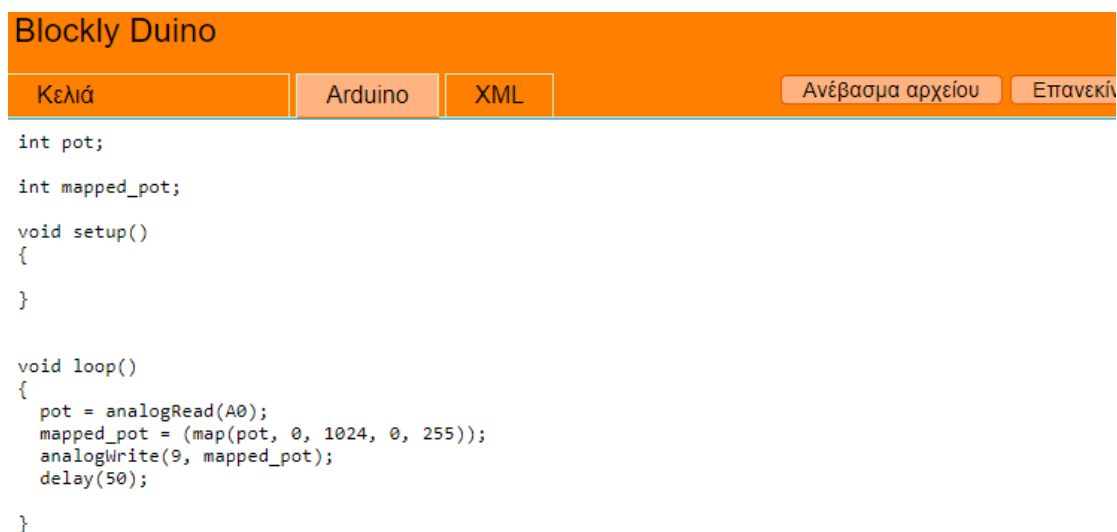
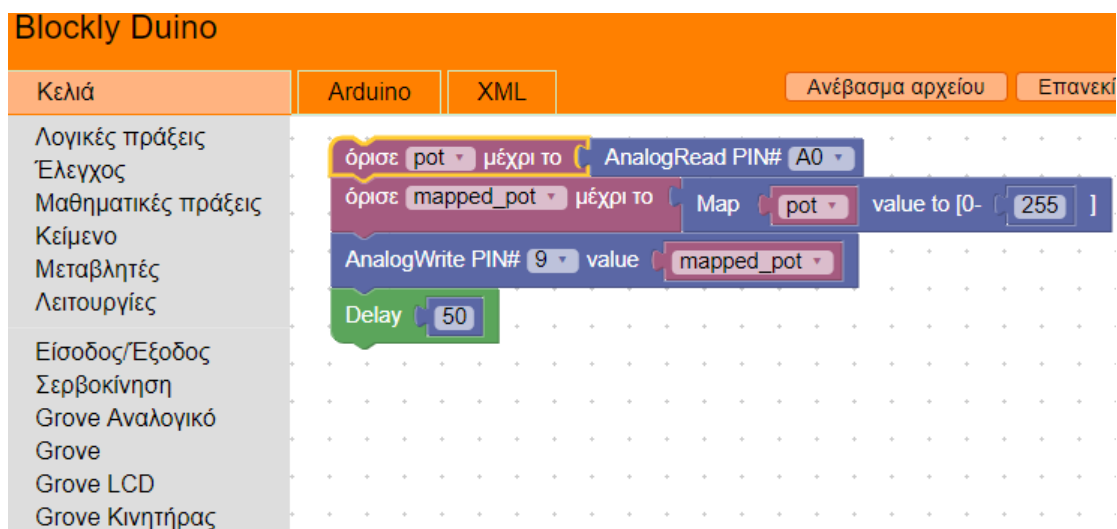


Εικόνα 26. Έλεγχος σερβοκινητήρα με ποτενσιόμετρο

Τους σερβοκινητήρες τους ελέγχουμε χρησιμοποιώντας παλμούς PWM από ένα Arduino. Το εύρος της συνάρτησης AnalogWrite είναι 0-255, οπότε πρέπει να μετατρέψουμε την ένδειξη του ποτενσιόμετρου στο εύρος τιμών PWM. Για αυτό χρησιμοποιούμε το block map, που βρίσκεται στην καρτέλα Μαθηματικές πράξεις.



Πρέπει να καθορίσουμε το αναμενόμενο εύρος εξόδου (0-255). Οπότε έχουμε:



Εικόνα 27. Block και κώδικας παραδείγματος 4

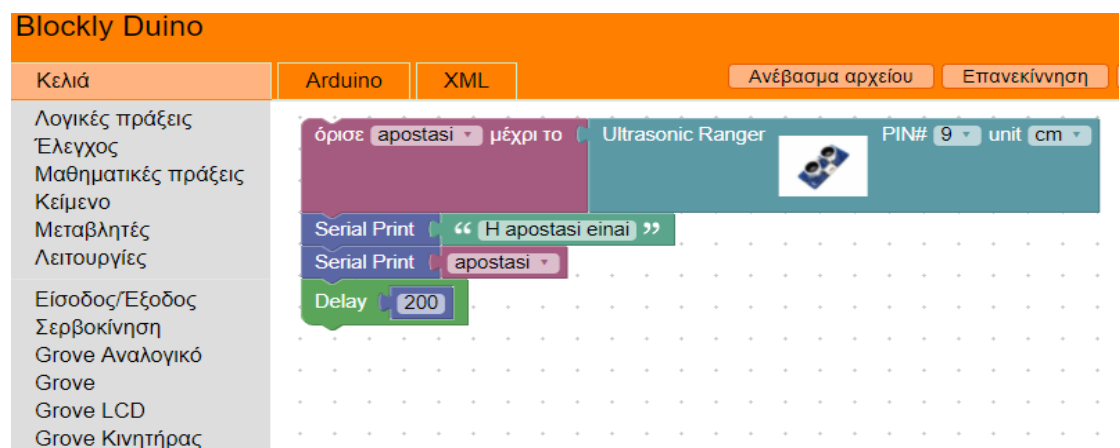
Βέβαια το BlocklyDuino διαθέτει διάφορα έτοιμα block περιφερειακών τα οποία μπορούμε να χρησιμοποιούμε ως αντικατάσταση κώδικα εφόσον τα έχουμε στη διάθεση μας. Ανήκουν στην σειρά Grove και είναι συμβατά με ειδική πλακέτα ([shield](#))

η οποία ενσωματώνεται στο Arduino. Για παράδειγμα το συγκεκριμένο πρόγραμμα θα μπορούσε να έχει μια εναλλακτική μορφή όπως παρακάτω.



Εικόνα 28. Εναλλακτική λύση παραδείγματος 4

Με παρόμοιο τρόπο μπορούμε να χρησιμοποιήσουμε και τον ανιχνευτή απόστασης (ultrasonic sensor). Η Serial Print εμφανίζει αρχικά στην οθόνη το μήνυμα «Η απόσταση είναι» και έπειτα την τιμή που ανιχνεύει ο αισθητήρας.



Έτσι απλά, μάθαμε για τα σήματα διαμόρφωσης πλάτους παλμού και πώς να εξάγουμε ένα σήμα PWM μέσω των εισόδων ενός Arduino χρησιμοποιώντας το μπλοκ AnalogWrite. Χρησιμοποιώντας τον βρόχο FOR, αλλάζουμε την τιμή PWM και ελέγχουμε την ένταση του LED.

Μάθαμε επίσης για το ποτενσιόμετρο και πώς να διαβάσουμε την αναλογική του τιμή χρησιμοποιώντας το μπλοκ AnalogRead. Στη συνέχεια παρακολουθήσαμε την τιμή του ποτενσιόμετρου χρησιμοποιώντας τον βρόχο WHILE και ενεργοποιήσαμε και απενεργοποιήσαμε ένα LED με βάση την ένδειξη κατωφλίου ποτενσιόμετρου.

Στην τελική άσκηση, χρησιμοποιήσαμε τα μπλοκ AnalogRead και AnalogWrite για να διαβάσουμε την τιμή POT και να ελέγξουμε έναν σερβοκινητήρα. Μάθαμε επίσης για τη λειτουργία χάρτη του Arduino και της χρήσης του και είδαμε εναλλακτικά παραδείγματα βάση των περιφερειακών που διαθέτει το BlocklyDuino από προεπιλογή.

7. Συμπεράσματα

7.1 Το block factory

Όσο χρησιμοποιούμε το BlocklyDuino, σε βάθος χρόνου μπορεί να μας φανεί εν μέρη ελλιπές. Όπως είχαμε αναφέρει, έχουμε στη διάθεση μας έναν μικρό αλλά περιεκτικό αριθμό blocks της σειράς grove, ικανών για την υλοποίηση προγραμμάτων χωρίς ιδιαίτερες απαιτήσεις περιφερειακών εξαρτημάτων.

Για τον λόγο αυτό υπάρχει το block factory της Google, μια πλατφόρμα που μας επιτρέπει να παράγουμε νέα blocks για τις ανάγκες των εργασιών μας. Σε μελλοντικό χρόνο θα μπορούσε να γίνει μια μελέτη εξ ολοκλήρου για το block factory καθώς αποτελεί μεγάλο κομμάτι στον προγραμματισμό με blocks. Μπορούμε να το επισκεφτούμε σε [αυτή την σελίδα](#), αλλά και να διαβάσουμε περισσότερα για αυτό, [εδώ](#).

7.2 Συμβατά υπολογιστικά συστήματα

Στο κεφάλαιο 5 μιλήσαμε για το Raspberry Pi και το λειτουργικό του σύστημα Raspbian. Αυτό βασίζεται σε μια έκδοση των Linux η οποία ονομάζεται Debian. Εφόσον μας εξυπηρετεί το περιβάλλον αυτό καταλήγουμε στο γεγονός, πως το BlocklyDuino μπορεί να λειτουργήσει σε οποιαδήποτε άλλη έκδοση Linux εμείς επιθυμούμε αλλά και σε διαφορετική συσκευή πέρα από το raspberry pi.

7.3 BlocklyDuino και σφάλματα

Μπορεί να συναντήσουμε όπως έχουμε πει και πιο πριν, διάφορα σφάλματα κατά την εκτέλεση του ανεβάσματος του κώδικα στο Arduino. Αυτό κυρίως θα οφείλεται σε κάποιο λάθος στο στάδιο επεξεργασίας των πηγαίων αρχείων, σε κάποιο πρόβλημα με την σύνδεση στο δίκτυο μας, ή στο γεγονός πως η πλακέτα που έχουμε στην διάθεση μας δεν είναι αυθεντική

8. Βιβλιογραφία

1. Σ. Μπουλταδάκης, Γ. Πατουλίδης, Ν. Ασημόπουλος. «Προγραμματισμός μικροελεγκτών για μηχανικούς», Εκδόσεις Τζιόλα.
2. Arduino [Ηλεκτρονική πηγή], Διαθέσιμο στον διαδικτυακό τόπο: en.wikipedia.org/wiki/Arduino. [Τελευταία πρόσβαση 05/10/2021].
3. Arduino Products [Ηλεκτρονική πηγή], Διαθέσιμο στο διαδικτυακό τόπο: www.arduino.cc/en/Main/Products. [Τελευταία πρόσβαση 10/09/2021].
4. Getting started with the Arduino IDE 2.0 [Ηλεκτρονική πηγή], Διαθέσιμο στον διαδικτυακό τόπο: www.arduino.cc/en/Tutorial/getting-started-with-ide-v2. [Τελευταία πρόσβαση 05/10/2021].
5. Scratch (programming language) [Ηλεκτρονική πηγή], Διαθέσιμο στον διαδικτυακό τόπο: [en.wikipedia.org/wiki/Scratch \(programming language\)](https://en.wikipedia.org/wiki/Scratch_(programming_language)). [Τελευταία πρόσβαση 05/10/2021].
6. Fotios Gioulekas, Panagiotis Katsaros. «Programming Arduino with Scratch (S4A)» Διαθέσιμο στον διαδικτυακό τόπο: <https://www.csd.auth.gr/images/ekdiloseis/STEMART.pdf>.
7. BlocklyDuino, Github [Ηλεκτρονική πηγή], Διαθέσιμο στον διαδικτυακό τόπο: <https://github.com/BlocklyDuino/BlocklyDuino>
8. Block factory για δημιουργία νέων μπλόκ στοιχείων [Ηλεκτρονική πηγή], Διαθέσιμο στον διαδικτυακό τόπο: <https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>