

ΑΓΓΕΛΙΚΗ ΜΠΕΛΕΧΑΚΗ



**ΕΚΠΑΙΔΕΥΤΙΚΗ
ΡΟΜΠΟΤΙΚΗ
ΜΕ ARDUINO**

ΑΘΗΝΑ 2021

Τίτλος:

Εκπαιδευτική Ρομποτική με Arduino

ISBN: 978-618-00-3211-6

Συγγραφέας: Αγγελική Γ. Μπελεχάκη

abele@sch.gr



Αυτό το υλικό διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού – Μη Εμπορική Χρήση – Παρόμοια Διανομή 4.0 (CC-BY-NC-SA)

Περιεχόμενα

ΕΙΣΑΓΩΓΗ	7
Εκπαίδευση STEM	7
STEM – Εκπαιδευτική Ρομποτική	8
Συστήματα εκπαιδευτικής ρομποτικής	9
Εκπαιδευτικές πλατφόρμες για την εφαρμογή της εκπαιδευτικής ρομποτικής	10
ARDUINO	11
Μικροελεγκτές	12
Περιγραφή του μικροελεγκτή Arduino	12
Είσοδοι / Έξοδοι	13
Ψηφιακά Pins	14
Αναλογικά Pins	14
Τροφοδοσία	15
Ενσωματωμένος διακόπτης	16
Ενσωματωμένα LED	16
Αισθητήρες	17
Υλικά	18
Breadboard	19
Προγραμματισμός ARDUINO	19
S4A Scratch for Arduino	19
Ardublock	20
Arduino IDE	21
Γλώσσα προγραμματισμού	22
Προγραμματισμός – Δομή προγράμματος	22
Μεταβλητές	23
Σχόλια	23

Βασικές συναρτήσεις	24
Συνάρτηση PinMode	24
Συνάρτηση digitalWrite	24
Συναρτήσεις digitalRead	25
Συνάρτηση analogWrite	25
Συνάρτηση analogRead	26
Βασικές εντολές	27
Δομή επιλογής	27
Εντολή switch	28
Δομή επανάληψης FOR	29
Βιβλιοθήκες	29
Σειριακή θύρα	30
ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ	32
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - LED που αναβοσβήνει	33
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Τρία LED που αναβοσβήνουν	36
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 - Μέτρηση Θερμοκρασίας Και Υγρασίας	39
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 – Συλλογή Τιμών Από Φωτοαντίσταση	42
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5 – Προγραμματισμός Κινητήρα Servo	45
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6 - Ποτενσιόμετρο και σχεδιογράφος processing	49
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7 - Κατασκευή ραντάρ με Arduino	57
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8 - Πρόγραμμα Που Κινεί Εναν Κινητήρα DC	64
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9 - Ρύθμιση Έντασης Led Με Αισθητήρα Bluetooth	69
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 10 – Πρόσβαση Ασφαλείας Με Αναγνώστη RFID	75
Βιβλιογραφία	83

ΠΡΟΛΟΓΟΣ

Το βιβλίο αυτό απευθύνεται σε όσους θέλουν να αποκτήσουν βασικές γνώσεις πάνω στη χρήση και τη λειτουργία του μικροελεγκτή Arduino.

Το Arduino είναι ένας μικροελεγκτής με πολλές εφαρμογές στην ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων. Χρησιμοποιείται και στην εκπαίδευση για τις δραστηριότητες STEM.

Στο πρώτο μέρος γίνεται μια αναφορά στο STEM και την εκπαιδευτική ρομποτική. Στη συνέχεια περιγράφεται ο μικροελεγκτής Arduino και τα χαρακτηριστικά του, τα εξαρτήματα που χρησιμοποιούνται για τα κυκλώματα και οι βασικές εντολές για το προγραμματισμό του.

Στο δεύτερο μέρος παρουσιάζονται δέκα φύλλα εργασίας με διάφορες κατασκευές κυκλωμάτων με το Arduino χρησιμοποιώντας τα πιο γνωστά εξαρτήματά του. Στα φύλλα εργασίας περιγράφονται τα κυκλώματα και δίνεται ο κώδικας γραμμένος στη γλώσσα του Arduino. Ακολουθούν ερωτήσεις κατανόησης για τους μαθητές.

Τα φύλλα εργασίας μπορούν να χρησιμοποιηθούν από εκπαιδευτικούς που θέλουν να εισάγουν τους μαθητές τους στη χρήση του Arduino ή και μαθητές που θέλουν να ασχοληθούν με αυτό.

Όλο τα προγράμματα που αναφέρονται στα φύλλα εργασίας βρίσκονται αναρτημένα στην διεύθυνση: <https://github.com/abele12/solar-charger>

Το βιβλίο έχει παιδαγωγικό σκοπό ο οποίος είναι να βοηθήσει τους μαθητές στην εκμάθηση του Arduino. Οι μαθητές με ευχάριστο και απλό τρόπο γνωρίζουν τα ψηφιακά κυκλώματα και τον προγραμματισμό τους και τέλος, με τη γνώση και την εμπειρία που παίρνουν από τα Φύλλα Εργασίας του βιβλίου μπορούν να δημιουργήσουν τις δικές τους σύνθετες εφαρμογές.

Τέλος, ευχαριστώ όσους με βοήθησαν με τις εύστοχες παρατηρήσεις τους ως προς τη δομή, τη σύνταξη και την καλλιτεχνική επιμέλεια. Κυρίως όμως ευχαριστώ την ομάδα ρομποτικής της Γ' τάξης του 2^{ου} Γυμνασίου Μελισσίων του σχολικού έτους 2019-20 που αποτέλεσε την πρώτη ομάδα με την οποία δοκιμάστηκε το υλικό και συμμετείχαμε με διάκριση στο 2^ο Πανελλήνιο Διαγωνισμό Ανοιχτών Τεχνολογιών στην Εκπαίδευση με το έργο: Δημιουργία ηλιακού φορτιστή με την πλατφόρμα Arduino (<https://github.com/abele12/solar-charger>).

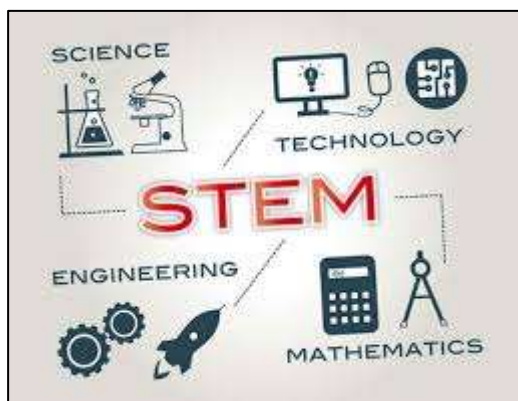
ΕΙΣΑΓΩΓΗ

Εκπαίδευση STEM

Ο όρος «εκπαίδευση STEM» αναφέρεται στη διδασκαλία και τη μάθηση στους τομείς της Επιστήμης, της Τεχνολογίας / Επιστήμης των Υπολογιστών, της Επιστήμης των Μηχανικών και των Μαθηματικών. Η εκπαίδευση στο STEM περιλαμβάνει διδακτικές-μαθησιακές ακολουθίες σε όλες τις βαθμίδες της εκπαίδευσης -από την προσχολική ηλικία έως το μεταδιδακτορικό επίπεδο τόσο κατά την διάρκεια του κανονικού ωρολογίου προγράμματος (π.χ. αίθουσες διδασκαλίας) όσο και κατά την διάρκεια δραστηριοτήτων που λαμβάνουν χώρα μετά την λήξη του κανονικού ωρολογίου προγράμματος (π.χ. προγράμματα σχολικών δραστηριοτήτων) (Gonzalez & Kuenzi, 2012).

Η υλοποίηση του STEM συνδυάζεται με την Υπολογιστική Επιστήμη (Computational Science), (Psycharis, 2016) ενώ ως μέθοδος επίλυσης προβλήματος χρησιμοποιείται η υπολογιστική σκέψη (computational thinking), (Wing,2006).

Σκοπός των μαθημάτων STEM είναι η εκμάθηση προγραμματισμού. Είναι μια μέθοδος πιο κατανοητή και ευχάριστη για τους μαθητές, διότι συνδυάζεται ο προγραμματισμός με τη δημιουργία κατασκευών με διάφορα συστήματα αλλά και με απλά υλικά όπως ξύλο, χαρτί κ.λπ. Χρησιμοποιείται η παιγνιώδης μάθηση με τα εργαλεία STEM και οι μαθητές μαθαίνουν χρησιμοποιώντας τη φαντασία, την πρόκληση, την περιέργεια και τον ανταγωνισμό (Κοτίνη & Τζελέπη, 2013). Δίνεται έμφαση στην αναλυτική σκέψη των μαθητών και γίνονται πιο δημιουργικοί. Τα παραδείγματα πρέπει να είναι φτιαγμένα λαμβάνοντας υπόψη το γνωστικό επίπεδο των μαθητών και με το χωρισμό των μαθητών σε ομάδες υποστηρίζεται ο μαθητής που δεν μπορεί να ανταπεξέλθει στη ροή του μαθήματος από τους υπόλοιπους της ομάδας με συζήτηση, συνεργασία και με τελικό σκοπό να καταλάβουν την άσκηση. Τέλος, συνδυάζονται όλα τα παραπάνω με τις γνώσεις που έχουν αποκτήσει από άλλα μαθήματα όπως τα Μαθηματικά, η Φυσική και η Τεχνολογία.



Εικόνα 1: Science – Technology – Engineering - Mathematics

1. STEM – Εκπαιδευτική Ρομποτική

Η εκπαιδευτική ρομποτική θεωρείται ως σχέδιο διεπιστημονικής δραστηριότητας κυρίως στην επιστήμη, τα μαθηματικά, την πληροφορική και την τεχνολογία, προσφέροντας σημαντικά νέα οφέλη σε όλα τα επίπεδα της εκπαίδευσης. Η εκπαιδευτική ρομποτική είναι μια ισχυρή και ευέλικτη διδασκαλία μάθησης που ενθαρρύνει τους/τις μαθητές/τριες να κατασκευάσουν και να ελέγξουν τα ρομπότ χρησιμοποιώντας συγκεκριμένες γλώσσες προγραμματισμού. Η Εκπαιδευτική Ρομποτική προάγει έναν ευχάριστο τρόπο μάθησης, ενώ παράλληλα προωθεί τα κίνητρα των παιδιών, την συνεργασία, την αυτοπεποίθηση και τη δημιουργικότητα. Πολλοί ερευνητές υποστηρίζουν ότι τα προγράμματα ρομποτικής παρέχουν μια πολύτιμη διαδρομή για την αύξηση του ενδιαφέροντος των παιδιών και τη συμμετοχή τους στην επιστήμη, την τεχνολογία, τη μηχανική και τα μαθηματικά (STEM), ενώ τα παρακινούν να ακολουθήσουν μια σταδιοδρομία σε έναν από αυτούς τους τομείς.

Η εκπαιδευτική ρομποτική αποτελεί ένα σχετικά νέο επιστημονικό κλάδο ο οποίος ασχολείται με την κατασκευή, τον προγραμματισμό και την αξιοποίηση των ρομπότ σε εκπαιδευτικό επίπεδο. Εμπεριέχεται ως διδακτικό αντικείμενο στο μάθημα της πληροφορικής και αποσκοπεί στο σχεδιασμό δραστηριοτήτων για την ενίσχυση δεξιοτήτων υπολογιστικής σκέψης (Ψαρά Ε., 2016). Τα τελευταία χρόνια υπάρχει μεγάλη απήχηση της εκπαιδευτικής ρομποτικής από τους εκπαιδευτικούς πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, εξαιτίας της αποτελεσματικότητας με την οποία βοηθά τους μαθητές στην απόκτηση δεξιοτήτων επίλυσης προβλημάτων (problem solving skills).

Για την επιτυχή ενσωμάτωση των εκπαιδευτικών ρομπότ στο σχολείο είναι απαραίτητο να υφίστανται δύο βασικοί παράγοντες:

- η ύπαρξη των ίδιων των ρομπότ ως κατασκευές και εργαλεία μάθησης στο σχολείο, τα οποία αποτελούν τα ηλεκτρονικά μέσα και,
- η δημιουργία και η κατάλληλη παιδαγωγική αξιοποίηση των ηλεκτρονικών υποδομών που έχουν τη δυνατότητα να υποστηρίξουν εκπαιδευτικές πλατφόρμες μάθησης, προγραμματισμού και αλληλεπίδρασης μεταξύ χρήστη / μαθητή και εκπαιδευτικού ρομπότ.

Τα οφέλη της εκπαιδευτικής ρομποτικής προς τους/τις μαθητές/τριες επικεντρώνονται στην αποτελεσματική οικοδόμηση της γνώσης (Κοντρουκτιβισμός) μέσω της ενεργούς συμμετοχής τους στις διαδικασίες σχεδιασμού και κατασκευής εκπαιδευτικών ρομπότ, γεγονός που σταδιακά οδηγεί σε εφαρμοσμένες πρακτικές και διαδικασίες επίλυσης προβλημάτων. Για να εφαρμοστεί το παραπάνω πλαίσιο είναι απαραίτητη η δημιουργία στοχευμένων δραστηριοτήτων που να εμπλέκουν τους μαθητές στον πειραματισμό με τα ρομπότ, ώστε να εξερευνήσουν τους κανόνες από τους οποίους διέπονται (Θεοχάρη Μ., 2013).

Η χρήση ειδικών κατασκευών τύπου ρομπότ για υλοποίηση μαθησιακών δραστηριοτήτων σε διάφορες γνωστικές περιοχές όπως η Φυσική, τα Μαθηματικά, η Πληροφορική, η Μηχανολογία και η Τεχνητή νοημοσύνη.



Εικόνα 2: Εκπαιδευτική Ρομποτική με Lego EV3

2. Συστήματα εκπαιδευτικής ρομποτικής

- 1970: επιδαπέδιες «χελώνες» – Πρόδρομος των Logo-like περιβαλλόντων – Πχ. Bigtrak, Roamer

- 1980: MIT+Lego κατασκευάζουν τον πρώτο προγραμματιζόμενο κύβο – Περιελάβανε μικροεπεξεργαστή που επέτρεπε την αποθήκευση προγράμματος και τον έλεγχο αισθητήρων και μηχανών
- Τέλος 1990: Lego Mindstorms NXT – Προγραμματιζόμενος κύβος NXT – Εκτέλεση κώδικα προγραμμάτων που αναπτύσσονται με διάφορα εργαλεία
- Πχ. RobotJ, JavaRobot, NXT – Έλεγχος αισθητήρων κίνησης, φωτός, ήχου, υγρασίας, κλπ.

3. Εκπαιδευτικές πλατφόρμες για την εφαρμογή της εκπαιδευτικής ρομποτικής

- Lego Mindstorms
- Arduino
- WeDo 2.0
- MakeBot
- Edison Robot
- Με διάφορα υλικά (π.χ. καλαμάκι, μπαλόνι, ξυλάκια, δίχαλα, χαρτόνι, πλαστικά συρίγγια, κινητήρες από χαλασμένα παιχνίδια, διάφορα εξαρτήματα από χαλασμένα παιχνίδια, ξυλάκια από παγωτό κλπ)



Εικόνα 3: Wedo



Εικόνα 4: Edison robot

ARDUINO

Τα μαθήματα με το Arduino ανήκουν στην κατηγορία: Εκπαίδευση με STEM. Το STEM (Science, Technology, Engineering, Mathematics) .

Το Arduino είναι ένας μικροελεγκτής με πολλές εφαρμογές στην ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων. Χρησιμοποιείται και στην εκπαίδευση για τις δραστηριότητες STEM.

Η κατασκευή του ξεκίνησε το 2005 με σκοπό να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο.

Οι ιδρυτές **Massimo Banzi και David Cueartielles** ονόμασαν το σχέδιο από τον Αρντουίνο της Ιβρέας και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

Το σχέδιο Arduino είναι μία διακλάδωση της υπολογιστικής πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Βασίζεται σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή, εισόδους και εξόδους, ψηφιακές και αναλογικές. Χρησιμοποιείται για τη δημιουργία πρωτοτύπων συσκευών αλλά και για εκπαιδευτικούς σκοπούς. Επάνω σε αυτή την πλατφόρμα μπορούν να συνδεθούν διαφόρων ειδών αισθητήρες θερμοκρασίας, πίεσης, φωτός, κινητήρες, LED, οθόνες κλπ.

Είναι ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και το λογισμικό και τα σχέδια που χρειάζονται για τη λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Συμπεριφέρεται σαν ένας μικρός υπολογιστής, αφού ο χρήστης μπορεί:

- να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και,
- να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου,
- να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

Το κύριο πλεονέκτημά του σε σχέση με άλλες διαδραστικές ηλεκτρονικές συσκευές είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και

επεκτείνει μια online γνωσιακή βάση και με εκτενή τεκμηρίωση που βοηθάει όποιον θέλει να δημιουργήσει έργο με αυτό

Ένα άλλο πλεονέκτημα του μικροελεγκτή είναι οι χαμηλές τιμές στις οποίες προσφέρεται τόσο ο μικροελεγκτής όσο και τα απαραίτητα εξαρτήματα.

1. Μικροελεγκτές

Μικροελεγκτής είναι ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα το οποίο διαθέτει επεξεργαστή, μνήμη, διάφορα περιφερειακά κυκλώματα καθώς επίσης και θύρες εισόδου/εξόδου για επικοινωνία με εξωτερικές συσκευές.
<http://www.eln.teilam.gr/sites/default/files/Lesson03.pdf>

Οι κατασκευαστικές εταιρείες κατασκεύασαν τους μικροελεγκτές ώστε ένα σύστημα να έχει μεγαλύτερες δυνατότητες και μικρότερο μέγεθος και έτσι ενσωμάτωσαν όλες τις λειτουργίες ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα.

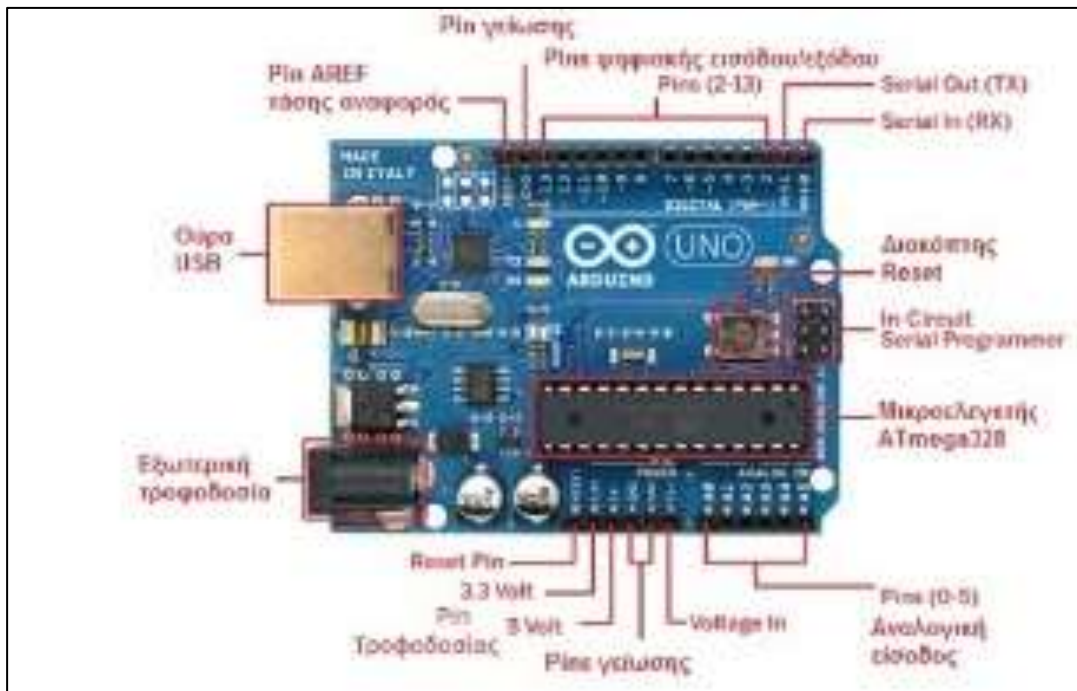
2. Περιγραφή του μικροελεγκτή Arduino

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM είναι η μνήμη για την αποθήκευση μεταβλητών των προγραμμάτων που φορτώνονται σε αυτό.
- 1Kb μνήμης EEPROM χρησιμοποιείται για την εγγραφή/ανάγνωση κατά την εκτέλεση των προγραμμάτων.
- 32Kb μνήμης Flash, όπου βρίσκεται το πρόγραμμα που έχει φορτωθεί μέσω της USB θύρας.

Και στις τρεις μνήμες δεν χάνονται τα περιεχόμενα όταν δεν υπάρχει τροφοδοσία ή αν γίνει RESET.

3. Είσοδοι / Έξοδοι



Εικόνα 5: Είσοδοι και έξοδοι της πλακέτας Arduino UNO

Το Arduino διαθέτει σειριακό interface και συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται:

- για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino και,
- για την αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί μέχρι 40mA.

3.1 Ψηφιακά pins

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Έτσι μπορεί να ανάψει ή να σβήσει ένα LED που έχει συνδεθεί στο συγκεκριμένο pin ή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin διοχετεύει ή όχι ρεύμα στο.

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 χρησιμοποιούνται για να σταλούν δεδομένα στη σειριακή μέσω της θύρας USB.
- Τα pin 2 και 3 μπορούν να ρυθμιστούν μέσα από το πρόγραμμά για να λειτουργήσουν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει και εκτελείται μια συγκεκριμένη συνάρτηση.
- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation). Συνδέοντας ένα LED σε κάποιο από αυτά τα pin μπορεί να ελεγχθεί πλήρως η φωτεινότητά του (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί την δυνατότητα δύο καταστάσεων (αναμμένο-σβηστό) που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Το PWM δεν είναι πραγματικά αναλογικό σύστημα και αν δώσουμε στην έξοδο την τιμή 127 δεν θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

3.2 Αναλογικά Pins

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που υπάρχει ενσωματωμένο στον μικροελεγκτή.

Αυτές οι θύρες χρησιμοποιούνται για την τροφοδοσία συσκευών με μια τάση η οποία μπορεί να μεταβάλλεται με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς V_{ref} η οποία, αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά μπορεί να διαβάσει την τιμή του pin ως ένα ακέραιο αριθμό, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V).

Για παράδειγμα, η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί το pin AREF με 3.3V και στην συνέχεια διαβαστεί κάποιο pin αναλογικής εισόδου που έχει εφαρμοστεί τάση 1.65V (= $3,3 / 2$), το Arduino θα επιστρέψει την τιμή 512 (= $1024 / 2$).

Επιπλέον, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή – που περιγράφονται στο κεφάλαιο Προγραμματισμός - μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0 έως 5 σε 14 έως 19 αντίστοιχα.

4. Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



Εικόνα 6: Τρόποι τροφοδοσίας Arduino

Η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V (ένας κοινός μετασχηματιστής του εμπορίου), από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το **πρώτο**, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) επανεκκνεί το Arduino.
- Το **δεύτερο**, με την ένδειξη 3.3V, τροφοδοτεί τα εξαρτήματά με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το **τρίτο**, με την ένδειξη 5V, τροφοδοτεί τα εξαρτήματά με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (η οποία λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το **τέταρτο** και το **πέμπτο** pin, με την ένδειξη GND, είναι γειώσεις.
- Το **έκτο** pin, (Vin) μπορεί να χρησιμοποιηθεί με δύο τρόπους. Ο πρώτος είναι μαζί με το pin γείωσης, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, αντί για την εξωτερική τροφοδοσία. Αν όμως υπάρχει συνδεδεμένη εξωτερική τροφοδοσία, μπορεί να χρησιμοποιηθεί για την τροφοδοσία συσκευών με την πλήρη τάση της εξωτερικής τροφοδοσίας από 7 έως 12V.

5. Ενσωματωμένος διακόπτης

Πάνω στην πλακέτα του Arduino υπάρχει ενσωματωμένος ένας διακόπτης micro-switch. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής.

6. Ενσωματωμένα LED

Στην πλακέτα Arduino υπάρχουν ενσωματωμένα 4 LED.

Τα δύο LED: TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface και ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω της USB.

Τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται μέσω των ψηφιακών pin 0 και 1.

Το LED με την σήμανση ON ανάβει όταν βρίσκεται σε λειτουργία.

Το LED με την σήμανση L είναι συνδεδεμένο με το ψηφιακό pin 13 και αν μέσα από το πρόγραμμα ανατεθεί η τιμή HIGH στο pin 13 τότε το ενσωματωμένο LED ανάβει .

7. Αισθητήρες

Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος (εμπόδιο) και παράγει από αυτό μία μετρήσιμη έξοδο.

Εφαρμογές: Οι αισθητήρες υπερήχων χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν λαμπρότερα ή απαλότερα αγγίζοντας τη βάση τους. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική. Η σημασία των αισθητήρων για τον άνθρωπο είναι σχεδόν αυτονόητη. Οι πρώτοι αισθητήρες εμφανίζονται μαζί με τα έμβια όντα που αποτελούν όργανα τους. Το μάτι και το αυτί είναι χαρακτηριστικό παράδειγμα, το πρώτο ανιχνεύει τμήμα του φάσματος της ηλεκτρομαγνητικής ακτινοβολίας και το δεύτερο τον ήχο, δηλαδή κύματα πίεσης.

Στο εμπόριο κυκλοφορούν πολλοί αισθητήρες για την πλακέτα Arduino και τη χρήση αρκετών από αυτούς περιγράφεται στα Φύλλα Εργασίας.



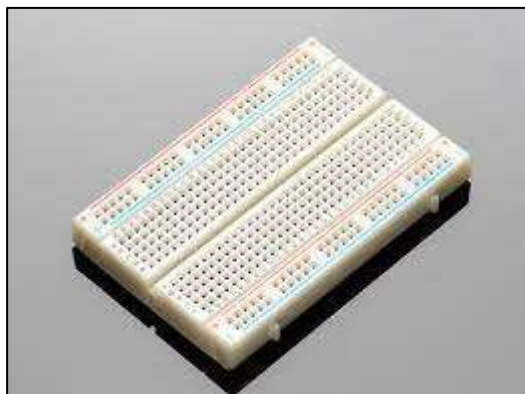
Εικόνα 7: Αισθητήρες και υλικά

8. Υλικά

Τα υλικά που διατίθενται στο εμπόριο είναι πολλά και αρκετά από αυτά θα τα παρουσιάσουν στα Φύλλα Εργασίας που ακολουθούν. Τα κυριότερα υλικά που διατίθενται για τη δημιουργία κυκλωμάτων με την πλακέτα Arduino είναι:

- Φωτοдиодοι (Led)
- Έγχρωμοι φωτοдиодοι (Rgb led)
- Κουμπιά (Buttons)
- Ποτενσιόμετρο (potentiometer)
- Ηχείο (Sounder)
- Αισθητήρες υπερήχων (ultrasonic)
- Αισθητήρας θερμοκρασίας
- Αισθητήρας bluetooth
- Φωτοαντιστάσεις (Photoresistor LDR)
- Σερβοκινητήρας (servo)
- DC Motor (κινητήρας συνεχούς κίνησης)

9. Breadboard



Εικόνα 8: Breadboard

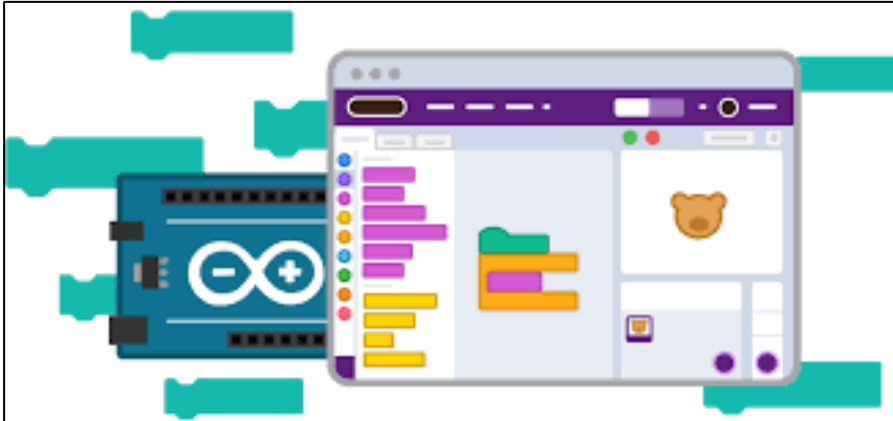
Το breadboard είναι μια πλαστική πλακέτα με υποδοχές οι οποίες συνδέονται μεταξύ τους με το αγώγιμο υλικό δημιουργώντας με αυτόν τον τρόπο κοινά σημεία. Η χρήση της πλακέτας αυτής διευκολύνει τη σύνδεση πολλών αισθητήρων στο Arduino, καθώς αυξάνεται σημαντικά ο αριθμός των υποδοχών.

10. Προγραμματισμός ARDUINO

Το Arduino μπορεί να προγραμματιστεί με τρία διαφορετικά περιβάλλοντα και τα οποία παρέχονται δωρεάν.

10.1 S4A Scratch for Arduino

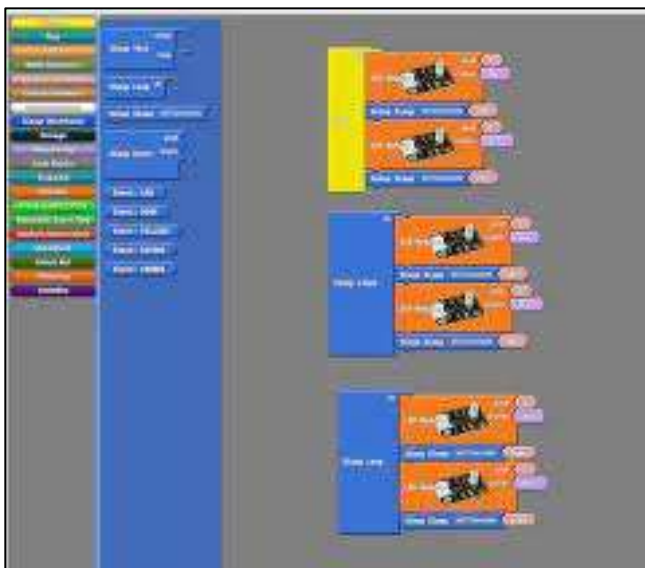
Το S4A είναι μια παραλλαγή του Scratch που επιτρέπει απλό προγραμματισμό της πλατφόρμας υλικού ανοιχτού κώδικα Arduino . Παρέχει μπλοκ για τη διαχείριση αισθητήρων και ενεργοποιητών που συνδέονται με το Arduino . Υπάρχει επίσης ένας πίνακας αναφοράς αισθητήρων.



Εικόνα 9: Περιβάλλον προγραμματισμού Scratch for Arduino

10.2 Ardublock

Το Ardublock είναι ένα οπτικό προγραμματιστικό περιβάλλον για την πλατφόρμα Arduino που χρησιμοποιεί όπως και το Scratch την τεχνική Σύρε και άσε (drag and drop) με έτοιμα blocks. Είναι ιδανικό για τη γνωριμία μικρών σε ηλικία μαθητών με το Arduino γιατί δεν πληκτρολογείται ο κώδικας ώστε να μην ασχολείται ο μαθητής με τη σωστή σύνταξη.



Εικόνα 10: Περιβάλλον προγραμματισμού Ardublock

10.3 **Arduino IDE** (<http://arduino.cc>)

Το Arduino IDE παρέχει:

- ένα περιβάλλον για την συγγραφή των προγραμμάτων σας (sketch),
- έτοιμα παραδείγματα,
- έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino,
- τον compiler για την μεταγλώττιση των sketch,
- ένα serial monitor (σειριακή οθόνη) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch σας
- και την επιλογή να ανεβάσετε το μεταγλωττισμένο sketch στο Arduino.

Το Arduino πρέπει να συνδεθεί σε μια από τις θύρες USB του υπολογιστή και να αναγνωριστεί από το λειτουργικό σας σύστημα ως εικονική σειριακή θύρα.

Η σύνδεση γίνεται ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών.

Για την αναγνώριση από το λειτουργικό πρέπει να εγκατασταθεί ο οδηγός του ελεγκτή Serial-over-USB ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE.

Όταν συνδεθεί, το κεντρικό παράθυρο του Arduino IDE θα εμφανιστεί όταν το εκτελέσετε και στο μενού Tools → Serial Port θα πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyusb## για το Linux). Επιλέξτε αυτή την εικονική θύρα και στην συνέχεια επιλέξτε τον τύπο του Arduino σας (Arduino Duemilanove w/ ATmega328) από το μενού Tools → Board.

Πλήρης τεκμηρίωση: <http://arduino.cc/en/Guide/HomePage>.



Εικόνα 11: Περιβάλλον προγραμματισμού Arduino IDE

11. Γλώσσα προγραμματισμού

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++.

11.1 Προγραμματισμός – Δομή προγράμματος

Σε ένα πρόγραμμα (sketch) υπάρχουν δύο βασικές συναρτήσεις:

Setup(): περιέχει όλες τις εντολές που εκτελούνται όταν ενεργοποιείται το Arduino (δηλαδή συνδέεται με το ρεύμα). Δίνονται οι αρχικές τιμές στις μεταβλητές που χρησιμοποιούνται στο πρόγραμμα και ορίζονται τα PINS που είναι συνδεδεμένες οι εισοδοί / έξοδοι. Αυτές οι εντολές εκτελούνται μια φορά στην αρχή.

Loop(): περιέχει τις εντολές που εκτελούνται στο κύριο πρόγραμμα και όταν φτάσει στο τέλος ενεργοποιείται ξανά μέχρι να συμβεί το γεγονός που έχουμε ορίσει για να σταματήσει το πρόγραμμα π.χ. να σταματήσει η τροφοδοσία ή να πατηθεί το πλήκτρο reset. Η εκτέλεση των εντολών ξεκινάει μόλις έχουν εκτελεστεί όλες οι εντολές της συνάρτησης setup.

Η δομή ενός προγράμματος είναι:

```
Void setup() {  
/*περιλαμβάνονται οι εντολές που αρχικοποιούν τις τιμές στο πρόγραμμα */  
}  
Void loop()  
/*οι εντολές που τρέχουν συνεχώς μέχρι να απενεργοποιηθεί το πρόγραμμα ή να πατηθεί το reset*/  
}
```

11.2 Μεταβλητές

Οι βασικοί τύποι μεταβλητών για τον προγραμματισμό του Arduino είναι:

Boolean, με τιμές 0 ή false και 1 ή True

Byte, ακέραιος με τιμές από 0 έως 255

Int, ακέραιος αριθμό από -32768 έως +32767

Long, ακέραιος αριθμό από -2147483648 έως +214748647

Float, δεκαδικοί αριθμοί

Char, ένας χαρακτήρας

String, πίνακας χαρακτήρων

Οι μεταβλητές δηλώνονται στη συνάρτηση setup() του προγράμματος.

Παράδειγμα:

int led = 13; όπου ορίζεται η ακέραια μεταβλητή led και παίρνει αρχική τιμή 13

float Val; όπου ορίζεται η πραγματική μεταβλητή Val

11.3 Σχόλια

Τα σχόλια που γράφονται σε μια γραμμή αρχίζουν με //.

Σε περίπτωση που τα σχόλια καταλαμβάνουν περισσότερες από μια γραμμές ξεκινούν με /* και στο τέλος */ ώστε ότι υπάρχει ανάμεσα σε αυτά τα σύμβολα αγνοείται από τον μεταγλωττιστή.

11.4 Βασικές Συναρτήσεις

i. Συνάρτηση pinMode

Περιγραφή: χαρακτηρίζει τα pins της πλακέτας σαν είσοδο ή έξοδο δηλαδή αν θα λάβει ή θα στείλει κάποιο σήμα.

Σύνταξη: pinMode(pin, Mode), όπου pin είναι ο αριθμός της θύρας και η Mode παίρνει την τιμή INPUT ή OUTPUT που χαρακτηρίζει την είσοδο ή την έξοδο.

Παράδειγμα: Εάν έχουμε κάποιο led συνδεδεμένο στο pin 2 και θέλουμε να στείλουμε σήμα να ανάψει ή να σβήσει, τότε το pin πρέπει να δηλωθεί ως έξοδος π.χ. pinMode(2,OUTPUT);.

Αντίστοιχα, εάν ένα κουμπί/διακόπτης είναι συνδεδεμένο στο pin 5, θέλουμε να λάβουμε ένα σήμα στο Arduino που όταν ενεργοποιηθεί για να κάνει κάποια λειτουργία και πρέπει να το δηλωθεί σαν είσοδος. π.χ. pinMode(5,INPUT);

Τα ψηφιακά pins παίρνουν τιμές από 0 έως 13 και τα αναλογικά από A0 έως A6. Για παράδειγμα:

```
pinMode(12, OUTPUT);
```

```
pinMode(led, OUTPUT);
```

```
pinMode(A2, INPUT);
```

ii. Συνάρτηση digitalWrite

Περιγραφή: Με αυτή τη συνάρτηση δηλώνουμε τι πρέπει να κάνει το Arduino με το pin.

Σύνταξη: digitalWrite(pin, Mode); όπου pin είναι ο αριθμός της θύρας και Mode είναι η τιμή της τάσης που θέλουμε να έχει το pin. Οι τιμές της Mode μπορεί να είναι LOW (0 V στο pin της εξόδου) ή HIGH (5 V στο pin της εξόδου).

Αν δώσουμε τη συνάρτηση: `digitalWrite(led, HIGH)`; τότε στην αντίστοιχη θύρα του led η τάση εξόδου είναι 5V. Για να γίνει αυτό πρέπει η αντίστοιχη θύρα να έχει οριστεί σαν θύρα εξόδου με την εντολή `pinMode(led, OUTPUT)`; Και αν δώσουμε την εντολή `val=digitalRead(led)`; Τότε η `val` θα πάρει την τιμή HIGH.

Παράδειγμα:

Αν θέλουμε να ανάψουμε ένα led χρησιμοποιούμε τη συνάρτηση `digitalWrite` ως εξής:

```
digitalWrite(2, HIGH);
```

Αν θέλουμε να το σβήσουμε τότε η συνάρτηση αλλάζει και γίνεται: `digitalWrite(2, LOW)`;

Αν δώσουμε τη συνάρτηση: `digitalWrite(led, HIGH)`; τότε στην αντίστοιχη θύρα του led η τάση εξόδου είναι 5V. Για να γίνει αυτό πρέπει η αντίστοιχη θύρα να έχει οριστεί σαν θύρα εξόδου με την εντολή `pinMode(led, OUTPUT)`;

iii. Συναρτήσεις `digitalRead`

Σύνταξη: `όνομα_μεταβλητής=digitalRead(pin)`; Δηλαδή επιστρέφει την τιμή της τάσης που δίνει το `pin`.

Περιγραφή: Με την `digitalRead` το Arduino να διαβάσει μια τιμή. Η τιμή αυτή και εδώ μπορεί να είναι HIGH ή LOW.

Παράδειγμα: Για παράδειγμα εάν έχουμε ένα κουμπί, το οποίο όταν τον πατάμε περνάει ρεύμα, τότε το διαβάζουμε ως εξής: `timi=digitalRead(5)`;

και η τιμή που θα πάρει θα είναι HIGH. Μόλις σταματήσουμε να τον πατάμε η `timi` θα γίνει LOW.

iv. Συνάρτηση `analogWrite`

Περιγραφή: Η συνάρτηση `analogWrite` χρησιμοποιείται για να δώσουμε τιμές από 0 έως 255. Η λειτουργία αυτή ονομάζεται Pulse Width Modulation (PWM). Λειτουργεί σαν διακόπτης που ανοιγοκλείνει πολύ γρήγορα στέλνοντας παλμούς 0 ή 255 και λόγω της ταχύτητας φαίνεται να στέλνει τιμές από 0 έως 255. Έτσι εάν θέλαμε να περιστρέψουμε ένα μοτέρ με την μισή ταχύτητα θα κάναμε χρήση της συνάρτησης ως εξής: `analogWrite(3,127)`;

Συμπερασματικά με τη συνάρτηση `digitalWrite` μπορεί να αποδοθεί έξοδος με μόνο δυο τιμές, HIGH και LOW. Με την `analogWrite` χρησιμοποιείται αν θέλουμε να χαμηλώσουμε τον φωτισμό σε ένα led ή να περιστρέψουμε ένα DC motor με χαμηλότερη ταχύτητα.

Σύνταξη: `analogWrite(pin, value);`

Με τιμές από 0 έως 255 προσομοιώνεται το διάστημα από 0 έως 5V με τη χρήση της συνάρτησης `analogWrite(Pin, Value)`, όπου το `pin` είναι ο αριθμός της θύρας για την οποία θα δώσουμε ρεύμα εξόδου και `Value` η τάση εξόδου που κυμαίνεται από 0 έως 5V που αναπαρίστανται αναλογικά.

Παράδειγμα: Μπορούμε να δώσουμε ενδιάμεσες τιμές ως εξής: `analogWrite(led, 127);` που αναλογεί σε τάση 2,5V,

Τη λειτουργία αυτή μπορούν να υποστηρίξουν μόνο τα PWM pins τα οποία είναι: 3, 5, 6, 9, 10, 11. Και η αντίστοιχη θύρα πρέπει να έχει οριστεί σαν εξόδου στη διαδικασία `setup()` με τη συνάρτηση `pinMode` ως εξής: `pinMode(10, OUTPUT);`

v. Συνάρτηση `analogRead`

Περιγραφή: Την συνάρτηση `analogRead` την χρησιμοποιούμε όταν θέλουμε να διαβάσουμε μια αναλογική τιμή, π.χ. από ένα αισθητήρα θερμοκρασίας, αισθητήρα υπερήχων, αισθητήρα υπερύθρων. Οι τιμές που μπορεί να διαβάσει είναι από 0 έως 1023.

Σύνταξη: `όνομα_μεταβλητής=analogRead(pin);`

Στο Arduino υπάρχουν 6 αναλογικές εισοδοί που συμβολίζονται με A0, A1, A2, A3, A4, A5. Μπορούμε να συνδέσουμε ένα αναλογικό εξάρτημα όπως το ποτενσιόμετρο και να το διαβάσουμε σαν είσοδο με τη συνάρτηση `analogRead(Pin)` όπου το `Pin` είναι ο αριθμός που αντιστοιχεί στη θύρα για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 έως 1023.

Παράδειγμα: `int r=analogRead(A1);`

Η αντίστοιχη θύρα πρέπει να οριστεί σαν εισόδου στη διαδικασία `setup()` με τη χρήση της συνάρτησης `pinMode` ως εξής: `pinMode(A1, INPUT);`

11.5 Βασικές εντολές

i. Δομή επιλογής

Σύνταξη: If <συνθήκη> {<μπλοκ εντολών 1>}

Else { <μπλοκεντολών 2>}

Όπου στη συνθήκη γίνεται ο έλεγχος που θέλουμε χρησιμοποιώντας τους τελεστές σύγκρισης <, >, =, <=, >= π.χ. pin_Val > 500.

Επίσης χρησιμοποιούνται οι λογικοί τελεστές:

||: για το Ή

&&: για το ΚΑΙ

Ένα παράδειγμα είναι: (pin_Val > 500) && (timer >= 1000).

Στα μπλοκ εντολών εκτελούνται οι εντολές “μπλοκ εντολών 1” αν ισχύει η συνθήκη διαφορετικά εκτελούνται οι εντολές “μπλοκ εντολών 2”. Το κομμάτι else μπορεί και να παραληφθεί.

Παράδειγμα:

```
if(distance<=120)
{
setColor(0,0,255);
buzzer2();
}
else
{
setColor(0,255,0);
buzzer3();
}
```

Σε αυτό το παράδειγμα αν η απόσταση που έχει μετρηθεί είναι μικρότερη ή ίση από 120 τότε ορίζεται με συνάρτηση setColor το χρώμα κίτρινο και παράγεται από το buzzer ένας ήχος, διαφορετικά ορίζεται το χρώμα πράσινο και παράγεται ένας διαφορετικός ήχος.

ii. Εντολή switch

Περιγραφή: Η εντολή switch ανήκει και αυτή στην κατηγορία εντολών «δομή επιλογής» και αντίθετα από την εντολή if, εδώ μπορούμε να επιλέξουμε από ένα σενεργειών. Μόνο η πρόταση που θα επιλεγεί εκτελείται ενώ όλες οι άλλες περιπτώσεις θα αγνοηθούν από το πρόγραμμα και αυτό θα συνεχίσει στην επόμενη εντολή.

Σύνταξη

```
switch (var) {  
  case label1:  
    // εντολές;  
    break;  
  case label2:  
    // εντολές;  
    break;  
  default:  
    // εντολές;  
    break;  
}
```

Παράδειγμα

```
switch(data)  
{  
  case '1': digitalWrite(13, HIGH);break;  
  case '0': digitalWrite(13, LOW);break;  
  default : break;  
}
```

Σε αυτό το παράδειγμα όταν η μεταβλητή data ;exei tim;h 1 το led ανάβει, όταν η μεταβλητή data έχει την τιμή 0 το led σβήνει και στις υπόλοιπες περιπτώσεις δεν συμβαίνει κάτι.

iii. Δομή επανάληψης

Η πιο συνηθισμένη εντολή επανάληψης είναι η **FOR**.

Περιγραφή: Αυτή η εντολή χρησιμοποιείται όταν θέλουμε να επαναλάβουμε κάποια διαδικασία πολλές φορές.

Σύνταξη:

For (<αρχική τιμή>; <συνθήκη τερματισμού>; <βήμα>)

{<εντολές>}

Στην αρχική τιμή δίνουμε την αρχική τιμή π.χ. $i=0$

Στο βήμα δίνουμε την αλλαγή της επανάληψης π.χ. $i+5$ (το $i+1$ μπορεί να παρασταθεί και σαν $i++$)

Και στη συνθήκη τερματισμού δίνουμε τη συνθήκη για να τελειώσει η επανάληψη π.χ. $i<10$, όπου όσο ισχύει αυτή η συνθήκη η εντολή εκτελείται.

Παράδειγμα:

```
For(i=1; i<10; i=i+1) {
  value = value + 1
  analogWrite(led, value);
};
```

Η εντολή for έχει προκαθορισμένο αριθμό βημάτων. Υπάρχουν άλλες εντολές που επαναλαμβάνονται όσο ισχύει μια συνθήκη.

- **While** <συνθήκη> {<εντολές>} δηλαδή όσο ισχύει η συνθήκη τρέχουν οι εντολές.
- **Repeat** {<εντολές>} until <συνθήκη> οι εντολές εκτελούνται όσο δεν ισχύει η συνθήκη.

11.6 Βιβλιοθήκες

Οι βιβλιοθήκες επεκτείνουν το περιβάλλον του Arduino και παρέχουν επιπλέον λειτουργικότητα στα προγράμματα. Οι περισσότεροι αισθητήρες για να προγραμματιστούν πρέπει να ενσωματώσουμε την αντίστοιχη βιβλιοθήκη.

Για να ενσωματωθεί μια έτοιμη βιβλιοθήκη πρέπει ο χρήστης να την κατεβάσει από το διαδίκτυο ή να δημιουργήσει τη δική του.

Η διαδικασία εισαγωγής μιας βιβλιοθήκης στο πρόγραμμα Arduino γίνεται με τη χρήση του Sketch στο μενού και μετά επιλέγεται Import Library (εισαγωγή βιβλιοθήκης). Εμφανίζεται η επιλογή Add Library (προσθήκη βιβλιοθήκης).



Εικόνα 12: Εισαγωγή βιβλιοθήκης στο Arduino IDE

11.7 Σειριακή θύρα

Το Arduino επικοινωνεί με τον υπολογιστή μέσω ενός καλωδίου USB, για να το τροφοδοτηθεί με ρεύμα και για να ανέβει το πρόγραμμα σε αυτό, αλλά και για να ανταλλάσσουμε δεδομένα μέσω σειριακής επικοινωνίας.

Μπορούμε να χρησιμοποιήσουμε την σειριακή οθόνη (Serial Monitor) που περιλαμβάνεται στο Arduino IDE ή να γράψουμε τα δικά μας προγράμματα τα οποία θα στέλνουν και θα δέχονται δεδομένα από την σειριακή θύρα για να χρησιμοποιηθούν είτε σε κάποιον αισθητήρα είτε σε κάποιο πρόγραμμα όπως το processing (παράδειγμα θα δούμε στο φύλλο εργασίας 6 και στο φύλλο εργασίας 9).

Η σειριακή οθόνη εμφανίζει τα σειριακά δεδομένα που στέλνονται από το Arduino. Μπορούμε να επιλέξουμε το ρυθμό μετάδοσης δεδομένων (baud rate) ανάλογα με την τιμή που θα επιλέξουμε στο πρόγραμμά μας με την εντολή `Serial.begin()`. Συνήθως χρησιμοποιείται η τιμή 9600 εξ ορισμού `<Serial.begin(9600);>`.

Για να δούμε τις τιμές που καταγράφει ο αισθητήρας θα χρησιμοποιήσουμε τη λειτουργία της παρακολούθησης της σειριακής από το περιβάλλον του προγράμματος.

Εντολές σειριακής θύρας

`Serial.begin(ρυθμός_μετάδοσης);` Η εντολή αυτή ενεργοποιεί τη σειριακή θύρα και έτσι μπορεί να εμφανιστεί η πληροφορία στο παράθυρο της σειριακής.

`Serial.println("<κείμενο>");` Η εντολή αυτή τυπώνει την πληροφορία στο παράθυρο της σειριακής και δημιουργεί μια νέα γραμμή. Παράδειγμα: Η εκτέλεση της εντολής `Serial.println("Καλημέρα!");` τυπώνει το λεκτικό Καλημέρα! στη σειριακή οθόνη ακολουθούμενη από το Enter.

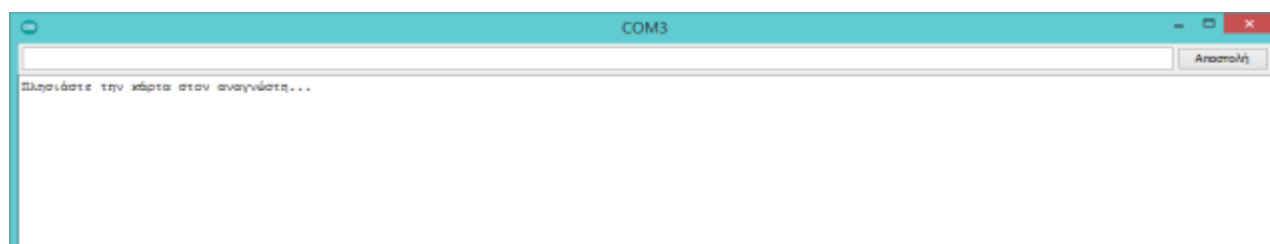
`Serial.print("<κείμενο>");` Η εντολή αυτή τυπώνει την πληροφορία στο παράθυρο της σειριακής. Παράδειγμα: Η εκτέλεση της εντολής `Serial.print("Καλημέρα!");` τυπώνει το λεκτικό Καλημέρα!.

`Serial.write();` : στέλνει δεδομένα στη σειριακή θύρα

`Serial.read();` : παραλαμβάνει δεδομένα από τη σειριακή θύρα

Παραδείγματα

- `serial.write(45);` ⇒ στέλνει την τιμή 45 στη σειριακή θύρα
- `Serial.write(val);` ⇒ στέλνει την τιμή της μεταβλητής `val` στη σειριακή θύρα
- `serial.write("hello");` ⇒ στέλνει τη λέξη hello στη σειριακή θύρα
- `serial.read(45);` ⇒ η σειριακή θύρα παραλαμβάνει την τιμή 45
- `serial.read(45);` ⇒ η σειριακή θύρα παραλαμβάνει την τιμή 45
- `serial.read(val);` ⇒ η η σειριακή θύρα παραλαμβάνει την τιμή της μεταβλητής `val`



Εικόνα 13: Η σειριακή οθόνη

ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - LED που αναβοσβήνει

Σε αυτή την άσκηση θα δημιουργήσουμε ένα κύκλωμα με ένα led που θα ανάβει και θα σβήνει ανά ένα δευτερόλεπτο.

Βαθμός Δυσκολίας: Εύκολο

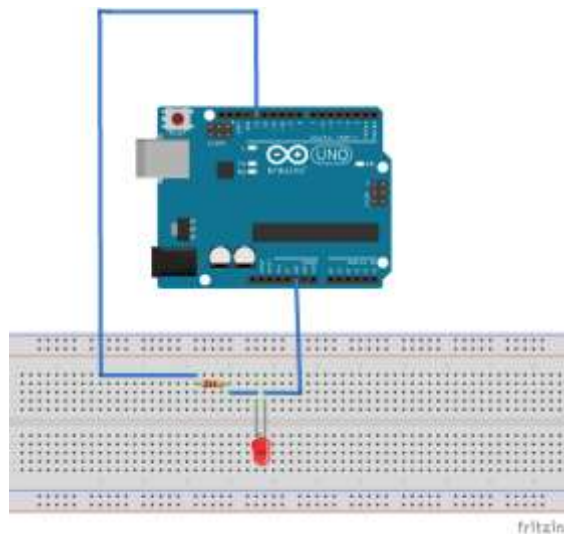
Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. Αντίσταση 220 Ω
3. LED
4. 2 jumper καλώδια
5. breadboard

Περιγραφή

Χρησιμοποιούμε το LED και μια αντίσταση ώστε το ρεύμα που περνάει, να αντιστοιχεί στην τάση λειτουργίας του LED

Το κύκλωμα φαίνεται στην φωτογραφία.



Η είσοδος του ρεύματος είναι στο PIN 13 και η αντίσταση συνδέεται ανάμεσα στην έξοδο και στο LED ώστε να ρυθμίζεται η ποσότητα ρεύματος που περνάει από αυτό. Το LED συνδέεται με το μακρύτερο πόδι στο pin 8 (φορά +). Το κύκλωμα κλείνει με το άλλο πόδι του LED στην υποδοχή GND (φορά -).

Αλγόριθμος

1. Δήλωση μεταβλητής που αντιστοιχεί στο PIN ίση με 13
2. Ορίζουμε το PIN 13 ως Έξοδο
3. Δίνουμε τάση 5V στο PIN 13
4. Περιμένουμε 1 δευτερόλεπτο με το LED αναμμένο
5. Διακόπτουμε το ρεύμα στο PIN 13
6. Περιμένουμε 1 δευτερόλεπτο με το LED σβηστό.
7. Τα βήματα 3 έως 6 επαναλαμβάνονται συνεχώς.

Πρόγραμμα

```

1. int led = 13;
2. void setup() {
3.   pinMode(led, OUTPUT);
4. }
5. void loop() {
6.   digitalWrite(led, HIGH);
7.   delay(1000);
8.   digitalWrite(led, LOW);
9.   delay(1000);
10. }

```

Οδηγίες

Στο φάκελο σας θα βρείτε το αρχείο led.ino. Κάνοντας διπλό κλικ πάνω στο αρχείο θα ανοίξει μέσα από το προγραμματιστικό περιβάλλον του Arduino. Μεταγλωττίστε και ανεβάστε το πρόγραμμα στην πλακέτα του Arduino αφού τη συνδέσετε στον υπολογιστή σας μέσω της USB θύρας.

Για τη μεταγλώττιση επιλέξτε το Μενού: Σχέδιο > Επικύρωση > Μεταγλώττιση και για το ανέβασμα του εκτελέσιμου κώδικα από το Μενού: Σχέδιο > Ανέβασμα.

Περιγράψτε τι κάνει το πρόγραμμα;

Ερωτήσεις

1. Σβήστε το ερωτηματικό από την εντολή digitalWrite(LED, HIGH); Μεταγλωττίστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε; Τι μήνυμα εμφανίζεται;
.....
Τι είδους λάθος δημιουργήσατε;
Διορθώστε το λάθος σας, εκτελέστε το πρόγραμμα.

2. Παρατηρήστε το Arduino και το Breadboard. Συμβουλευτείτε και τον κώδικα του προγράμματος. Σε ποιον αποδέκτη (pin) είναι συνδεδεμένο το LED;
.....
3. Ποια είναι η εντολή με την οποία ανάβει το LED;
.....
4. Με ποια είναι η εντολή σβήνει το LED;
5. Αλλάξτε την εντολή `digitalWrite(LED, HIGH);` με την `digitalWrite(13, HIGH);` και μεταγλωττίστε το πρόγραμμα. Τι συμβαίνει; Δικαιολογήστε την απάντησή σας.
.....
6. Αλλάξτε το καλώδιο που συνδέει το LED από pin 13 στο pin 10. Τροποποιήστε το πρόγραμμα ώστε το LED να αναβοσβήνει.
7. Η εντολή `delay(1000);` δημιουργεί μια καθυστέρηση 1000 msec = 1. sec. Κάντε τις κατάλληλες τροποποιήσεις στο πρόγραμμα έτσι ώστε το LED να ανάβει για 2 sec και να σβήνει για 3 sec. Δοκιμάστε το πρόγραμμα.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Τρία LED που αναβοσβήνουν

Σε αυτήν την άσκηση θα συνδέσουμε τρία led στο Arduino τα οποία θα αναβοσβήνουν ανά 3 δευτερόλεπτα.

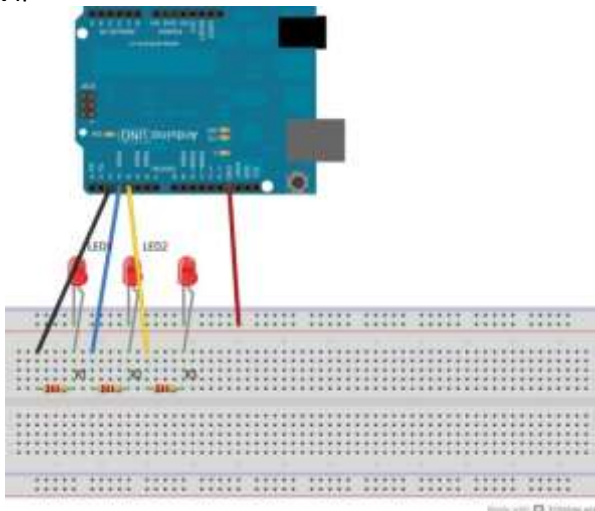
Βαθμός δυσκολίας: Εύκολο

Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. τρεις αντιστάσεις 220 Ω
3. τρία LED
4. 6 jumper καλώδια
5. breadboard

Περιγραφή

Στο κύκλωμα του Φύλλου Εργασίας 1 προσθέστε δύο ακόμη led όπως φαίνεται στο πιο κάτω σχήμα.



ΑΛΓΟΡΙΘΜΟΣ

1. Δίνουμε τάση 5V στο PIN 2
2. Περιμένουμε 3 δευτερόλεπτα με το LED αναμμένο
3. Διακόπτουμε το ρεύμα στο PIN 2
4. Επαναλαμβάνουμε τα βήματα 1 έως 3 για το pin 3 και μετά για το pin 4 και στη συνέχεια όλα τα πιο πάνω βήματα επαναλαμβάνονται συνεχώς .

Πρόγραμμα

Προσθέστε στο πρόγραμμα τις κατάλληλες εντολές ώστε τα τρία led να ανάβουν διαδοχικά ανά τρία δευτερόλεπτα.

Υπόδειξη

1. Void loop() {
2. digitalWrite(2, HIGH);
3. delay(3000);
4. digitalWrite(2, LOW);
5. digitalWrite(3, HIGH);
6. delay(3000);
7. digitalWrite(...);
8. digitalWrite(4, HIGH);
9. delay(3000);
10. digitalWrite(...);
11. }

Οδηγίες

1. Μεταγλωττίστε το πρόγραμμα και ανεβάστε το στην πλακέτα Arduino,
2. Τροποποιείστε το πρόγραμμα ώστε το πρώτο LED να ανάβει για 3 δευτερόλεπτα, μετά να σβήνει, μετά το δεύτερο να ανάβει για 2 δευτερόλεπτα και μετά να σβήνει για 1 δευτερόλεπτο και στη συνέχεια να ανάβουν και τα τρία για 3 δευτερόλεπτα και μετά να σβήνουν.
3. Αποθηκεύστε το πρόγραμμά σας με το όνομα flash_led.ino.

Ερωτήσεις

1. Περιγράψτε τι κάνει το πρόγραμμα

.....
.....
.....

Κάντε τις κατάλληλες αλλαγές στο πρόγραμμά σας ώστε το πρώτο LED να ανάβει για 5 δευτερόλεπτα, το δεύτερο για 3 και το τρίτο για 1.

2. Δοκιμάστε το πρόγραμμά σας.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 - Μέτρηση θερμοκρασίας και υγρασίας

Σε αυτήν την άσκηση θα χρησιμοποιήσουμε έναν αισθητήρα θερμοκρασίας και υγρασίας.

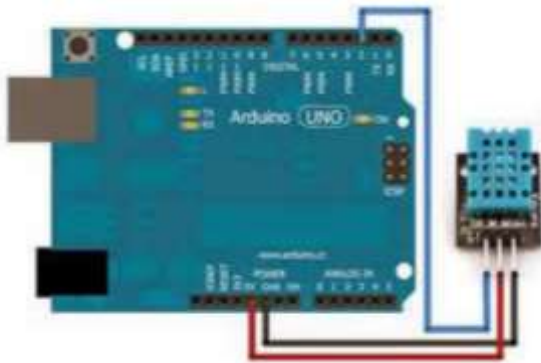
Βαθμός δυσκολίας: Εύκολο

Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. Αισθητήρας θερμοκρασίας και υγρασίας τύπου DHT11
3. Breadboard (μπορεί να μη χρησιμοποιηθεί και ο αισθητήρας να συνδεθεί απευθείας με την πλακέτα όπως φαίνεται στο σχήμα)

Περιγραφή κυκλώματος

Ο αισθητήρας συνδέεται στο κύκλωμα όπως φαίνεται στο πιο κάτω σχήμα. Για να λειτουργήσει ο αισθητήρας πρέπει να χρησιμοποιηθούν εντολές που βρίσκονται σε μια βιβλιοθήκη (DHT), η οποία πρέπει να ενσωματωθεί στο πρόγραμμα.



Σύνδεση με το Arduino

Pin: Μπλε καλώδιο

Ground: Μαύρο καλώδιο

5V: Κόκκινο καλώδιο

ΑΛΓΟΡΙΘΜΟΣ

1. Διαβάζουμε την τιμή από τον αισθητήρα
2. Τυπώνουμε στην σειριακή οθόνη την τιμή της υγρασίας
3. Τυπώνουμε στην σειριακή οθόνη την τιμή της θερμοκρασίας

4. Περιμένουμε 5 δευτερόλεπτα
5. Επαναλαμβάνουμε τα βήματα 1 έως 4

Βιβλιοθήκη

Για να λειτουργήσει ο αισθητήρας με το Arduino πρέπει να ενσωματωθεί η βιβλιοθήκη DHT στο πρόγραμμα όπως αναφέρεται στην εισαγωγή.

Εντολές

DHT.read11(dht_apin); Διαβάζεται η τιμή που αποδίδει ο αισθητήρας

Serial.print(DHT.temperature); Τυπώνεται η τιμή της θερμοκρασίας στο παράθυρο της σειριακής

Serial.print(DHT.humidity); Τυπώνεται η τιμή της υγρασίας στο παράθυρο της σειριακής

Πρόγραμμα

1. #include "dht.h"
2. dht DHT
3. void setup() {
4. Serial.begin(9600);
5. delay(500);
6. Serial.println("DHT11 Αισθητήρας Θερμοκρασίας και υγρασίας\n");
7. delay(1000);
8. }
9. void loop() {
10. DHT.read11(dht_apin); //διαβάζεται η τιμή από τον αισθητήρα
11. Serial.print("Υγρασία= ")
12. Serial.print(DHT.humidity); //τυπώνει την τιμή της υγρασίας
13. Serial.println("%");
14. Serial.print("Θερμοκρασία= ")
15. Serial.print(DHT.temperature); //τυπώνει την τιμή της θερμοκρασίας
16. Serial.println("°C");
17. delay(5000); // Αναμονή πέντε δευτερολέπτων για νέα πρόσβαση στον αισθητήρα
18. }

Οδηγίες

1. Ανοίξτε το temp_hum.ino, μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino
2. Το πρόγραμμα διαβάζει και εμφανίζει τις τιμές που δίνει ο αισθητήρας, Για να δείτε τις τιμές αυτές επιλέξτε από το Μενού: Εργαλεία>Παρακολούθηση Σειριακής.

Ερωτήσεις

1. Με ποια εντολή διαβάζουμε τις τιμές που δίνει ο αισθητήρας;
.....
2. Ποια μεταβλητή μας δίνει τις τιμές της υγρασίας;
.....
3. Ποια μεταβλητή μας δίνει τις τιμές της θερμοκρασίας
.....
4. Με ποια εντολή μπορούμε να δούμε την τιμή της θερμοκρασίας στη σειριακή οθόνη
.....

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 – Συλλογή τιμών από φωτοαντίσταση

Σε αυτό το μάθημα θα χρησιμοποιήσουμε μια φωτοαντίσταση (LDR – Light Dependent Resistor) που θα χρησιμοποιηθεί σαν αισθητήρας φωτός, ένα LED και ένα piezo Buzzer (ηχείο).

Βαθμός δυσκολίας: Εύκολο

Εφαρμογές

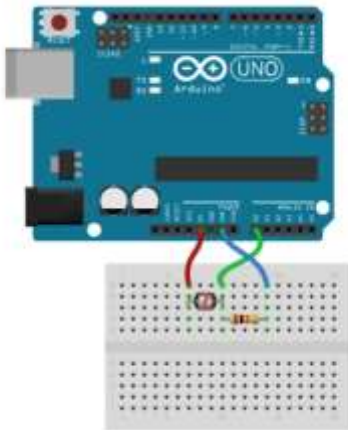
Η φωτοαντίσταση χρησιμοποιείται στα ηλεκτρονικά κυκλώματα φωτισμού σε περιπτώσεις που θέλουμε να ελέγξουμε το φωτισμό σε δρόμους ή σε ένα δωμάτιο.

Χρησιμοποιείται ακόμα σε αυτοματισμούς, συστήματα συναγερμού, πόρτες που ανοίγουν αυτόματα κ.λπ.

Περιγραφή κυκλώματος

Ο αισθητήρας φωτός (φωτοαντίσταση) λειτουργεί ως εξής: ανάλογα με την ένταση του φωτός που πέφτει πάνω του, περνάει ανάλογη ποσότητα ρεύματος από το κύκλωμα. Διαβάζοντας την ένδειξη που δίνει το pin πάνω στο οποίο βρίσκεται η φωτοαντίσταση όταν η τιμή είναι υψηλή, ο LDR δέχεται πολύ φως και χαμηλή όταν δεν υπάρχει φως.

Συνδέστε στο κύκλωμα του Arduino τη φωτοαντίσταση όπως φαίνεται στο πιο κάτω σχήμα. Η θύρα που συνδέεται η φωτοαντίσταση πρέπει να είναι αναλογική (εδώ είναι στο A0).



Αλγόριθμος

1. Διαβάζουμε την τιμή της έντασης του φωτός από τη φωτοαντίσταση
2. Αποθηκεύουμε την τιμή στη μεταβλητή R1
3. Τυπώνουμε την τιμή της R1 στην σειριακή οθόνη
4. Περιμένουμε 2 δευτερόλεπτα
5. Επαναλαμβάνουμε τα βήματα 1 έως 4

Πρόγραμμα

- ```
1. int LDR1 = A0; //συνδέεται η LDR1 στο Pin A0
2. void setup() {
3. Serial.begin(9600);
4. pinMode(LDR, INPUT);
5. delay(2000); }
6. void loop() {
7. int R1 = analogRead(LDR); // Διαβάζει την τιμή της φωτοαντίστασης με όνομα
 LDR 1
8. Serial.print("R1 = ");
9. Serial.println(R1);
10. delay(2000);
11. }
```

### ΟΔΗΓΙΕΣ

1. Ανοίξτε το αρχείο ldr.ino
2. Μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino

3. Το πρόγραμμα διαβάζει και εμφανίζει τις τιμές που δίνει η φωτοαντίσταση, Για να δείτε τις τιμές αυτές επιλέξτε από το Μενού: Εργαλεία > Παρακολούθηση Σειριακής.

### Ερωτήσεις

1. Περιγράψτε τη λειτουργία της εντολής `analogRead(LDR1)`
2. Ποια η λειτουργία της εντολής `pinMode(OUTPUT)`;
3. Ποιος ο ρόλος της μεταβλητής `R1`;
4. Τι τιμές δίνει ο αισθητήρας:

|                       |  |
|-----------------------|--|
| Στο φως;              |  |
| Όταν σκιάσετε λίγο;   |  |
| Όταν σκιάσετε αρκετά; |  |
| Στο σκοτάδι;          |  |

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5 – Προγραμματισμός Κινητήρα Servo

**Βαθμός Δυσκολίας:** Μέτριας δυσκολίας

### Περιγραφή κυκλώματος

Οι κινητήρες servo είναι μικρές συσκευές με έναν εξωτερικό άξονα. Αυτός ο άξονας περιστρέφεται σε διάφορες θέσεις όταν αποσταλεί στον servo ένα σήμα. Όσο υπάρχει το σήμα στην γραμμή εισόδου του Servo, αυτός θα διατηρεί τον άξονα του σε μια συγκεκριμένη θέση. Όταν αλλάζει το σήμα προκαλεί το servo να μεταβάλει την γωνία του άξονα.

**Εφαρμογές:** τα servo χρησιμοποιούνται σε τηλεχειριζόμενα αεροπλάνα, αυτοκίνητα και στην ρομποτική. Οι κινητήρες αυτοί έχουν ένα εξαιρετικά μικρό μέγεθος αλλά είναι αρκετά ισχυροί για το μέγεθος τους. Αυτό τους κάνει πολύ εύχρηστους.

### Σύνδεση με το Arduino

- Κόκκινο καλώδιο: 5V
- Πορτοκαλί: pin 9
- Μαύρο: ground

### Αλγόριθμος

Ορίζουμε το τη σύνδεση του servo στο pin 9.

1. Στρέφουμε τον κινητήρα 15 μοίρες
2. Περιμένουμε 1 λεπτό
3. Στρέφουμε τον κινητήρα 30 μοίρες
4. Περιμένουμε 1 λεπτό
5. Τα βήματα 2 έως 5 επαναλαμβάνονται ώστε ο κινητήρας να στρέφεται σε κάθε βήμα 15 μοίρες περισσότερο από το προηγούμενο μέχρι τις 90 μοίρες και στη συνέχεια να αλλάζει φορά και να μειώνει κατά 15 μοίρες τη στροφή.
6. Με την επαναληπτική εντολή for ο κινητήρας στρέφεται μια μοίρα κάθε 15 δευτερόλεπτα μέχρι τις 180 μοίρες και στη συνέχεια επιστρέφει στην αρχική του θέση με μια εντολή for και 15 δευτερόλεπτα ανάμεσα σε κάθε κίνηση.

## Βιβλιοθήκη

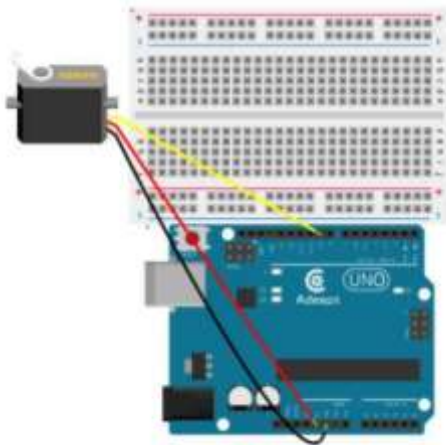
Η βιβλιοθήκη που υποστηρίζει τον σερβοκινητήρα στην επικοινωνία με το Arduino χρησιμοποιείται με την εντολή:

```
#include <Servo.h>
```

## Μέθοδοι

- attach(): ορίζει το pin που είναι συνδεδεμένος ο κινητήρας
- write(): ορίζονται οι μοίρες που στρέφεται ο κινητήρας

## Πρόγραμμα



```
/*Ο σερβοκινητήρας γυρίζει 15 μοίρες, 30 μοίρες, 45 μοίρες, 60 μοίρες, 75 μοίρες, 90
μοίρες, 75 μοίρες, 60 μοίρες, 45 μοίρες, 30 μοίρες, 15 μοίρες, 0 μοίρες και μετά 0 μοίρες
έως 180 μοίρες και από 180 μοίρες σε 0 μοίρες.*/*
```

1. #include <Servo.h>
2. Servo myservo;//δημιουργεί το αντικείμενο myservo
3. void setup(){
4. myservo.attach(9);//συνδέει τον κινητήρα στο pin 9
5. myservo.write(0);// ο άξονας του κινητήρα τοποθετείται στις 0 μοίρες
6. delay(1000);
7. void loop(){

```
8. myservo.write(15);//ο άξονας του κινητήρα τοποθετείται στις 15 μοίρες
9. delay(1000);
10. myservo.write(30);
11. delay(1000);
12. myservo.write(45);
13. delay(1000);
14. myservo.write(60);
15. delay(1000);
16. myservo.write(75);
17. delay(1000);
18. myservo.write(90);
19. delay(1000);
20. myservo.write(75);
21. delay(1000);
22. myservo.write(60);
23. delay(1000);
24. myservo.write(45);
25. delay(1000);
26. myservo.write(30);
27. delay(1000);
28. myservo.write(15);
29. delay(1000);
30. myservo.write(0);
31. delay(1000);
32. for(int num=0;num<=180;num++)
33. {
34. myservo.write(num);
35. delay(10);
36. }
37. for(int num=180;num>=0;num--)
38. {
39. myservo.write(num);
40. delay(10);
41. }
42. }
```

### Οδηγίες

1. Συνδέστε τον κινητήρα με το Arduino όπως φαίνεται στο πιο πάνω σχήμα.
2. Ανοίξτε το πρόγραμμα `servo.ino` και μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino

### Ερωτήσεις

1. Τι αποτέλεσμα έχει η εντολή `myservo.write(30);`  
.....  
.....  
.....
2. Ποια η χρήση της εντολής `attach();`  
.....  
.....  
.....
3. Πως ονομάζεται ο σερβοκινητήρας στο πρόγραμμα; .....
4. Ποια είναι η τιμή της μεταβλητής `num` στην εντολή `for(int num=0;num<=180; num++)` .....  
Πότε σταματάει να εκτελείται η εντολή `for`  
.....  
Πόσες φορές εκτελείται η εντολή `myservo.write(num);` .....
5. Αλλάξτε τις εντολές του προγράμματος των γραμμών 8 έως 19 με ένα μπλοκ εντολών `for`.
6. Αλλάξτε τις πιο κάτω εντολές του προγράμματος των γραμμών 20 έως 31 με ένα μπλοκ εντολών `for`.



## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6 - Ποτενσιόμετρο και σχεδιογράφος processing

**Βαθμός Δυσκολίας:** Δύσκολο

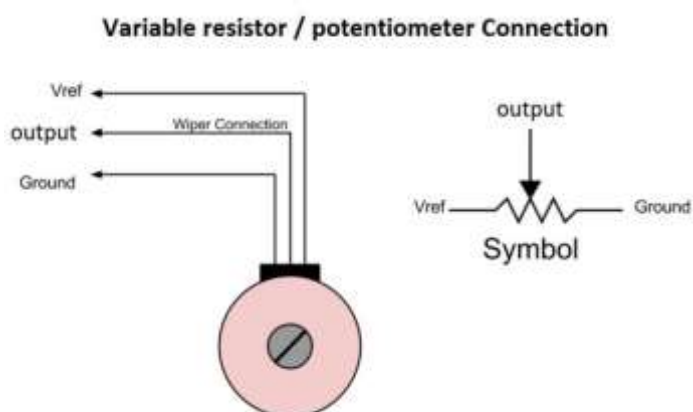
Σε αυτό το μάθημα θα συνδέσουμε ένα ποτενσιόμετρο με το Arduino και αλλάζοντας την τιμή της τάσης, αυτή θα απεικονίζεται σε μια οθόνη LCD 1602 και θα αποτυπώνεται γραφικά στην οθόνη με έναν κύκλο που αλλάζει το μέγεθός του ανάλογα με την τιμή της τάσης. Αυτό γίνεται με τη χρήση του προγράμματος processing χρησιμοποιώντας τη σειριακή θύρα του Arduino.

### Υλικά

- ποτενσιόμετρο
- οθόνη LCD 1602
- Arduino UNO
- Breadboard
- Διάφορα καλώδια σύνδεσης ηλεκτρονικών στοιχείων στο Breadboard
- Καλώδιο USB για τη σύνδεση του Arduino στον υπολογιστή

### Ποτενσιόμετρο

Τα ποτενσιόμετρα είναι μεταβλητές αντιστάσεις των οποίων η τιμή της αντίστασης μπορεί να αλλάξει γυρνώντας τον μοχλό που διαθέτουν στο πάνω μέρος τους. Σίγουρα έχετε χρησιμοποιήσει ένα στο παρελθόν, π.χ. για να αλλάξετε την ένταση σε ένα στερεοφωνικό.

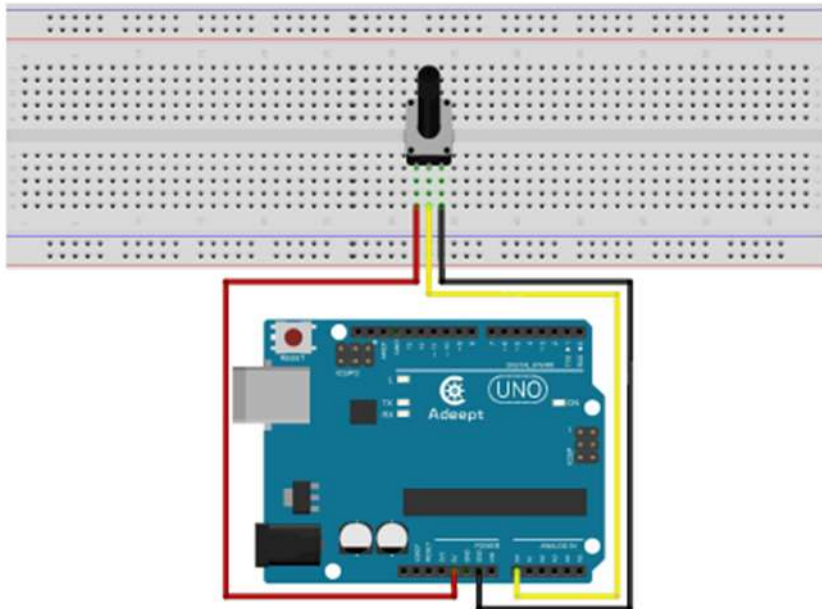


Τα ποτενσιόμετρα υπάρχουν σε διάφορα μεγέθη αντίστασης. Μπορούν να δώσουν τιμές

από 0 Ω μέχρι την μέγιστη τιμή για την οποία έχουν σχεδιαστεί. Για παράδειγμα, ένα ποτενσιόμετρο 10 KΩ μπορεί να δώσει τιμές από 0 Ω μέχρι μέγιστο 10 KΩ.

Όλα τα ποτενσιόμετρα έχουν τρεις ακροδέκτες. Οι εξωτερικοί χρησιμοποιούνται για την τροφοδοσία (Vred & GND). Ο μεσαίος ακροδέκτης αποδίδει την τιμή της μεταβλητής αντίστασης ανάλογα με την θέση του μοχλού.

Για να λειτουργήσει το ποτενσιόμετρο με το Arduino χρειάζεται η βιβλιοθήκη WIRE που ενεργοποιείται με την εντολή: `#include <Wire.h>`



Συνδέουμε τα τρία καλώδια από το ποτενσιόμετρο στο breadboard του Arduino.

- Το πρώτο καλώδιο συνδέεται στη γείωση
- Το δεύτερο (μεσαίο) συνδέει το αναλογικό pin A0 του Arduino
- Το τρίτο συνδέεται με τα 5V στο Arduino.

### Η οθόνη LCD 1602

Χρησιμοποιείται μια οθόνη LCD με 2 γραμμές των 16 χαρακτήρων. Σε αυτού του τύπου τις οθόνες μπορούμε να αποτυπώσουμε μόνο λατινικούς χαρακτήρες, σύμβολα και αριθμητικά ψηφία και όχι ελληνικούς χαρακτήρες.

Η οθόνη αποτυπώνει τα αποτελέσματα σε έναν πίνακα 16 στηλών και 2 γραμμών.

### Απεικόνιση γραφικών με τη γλώσσα processing

Για να απεικονίσουμε γραφικά τις τιμές που δίνει το ποτενσιόμετρο χρησιμοποιούμε πρόγραμμα γραφικών σε γλώσσα processing. Το πρόγραμμα processing είναι ένα

δωρεάν λογισμικό και μπορεί να ανακτηθεί από τη διεύθυνση: <https://processing.org/download>. Αυτό το πρόγραμμα μπορεί και επικοινωνεί με το Arduino μέσω της σειριακής θύρας με την εντολή `Serial.read()`. Η εντολή αυτή διαβάζει την τιμή της τάσης από τη σειριακή θύρα, Στη γλώσσα του Arduino χρησιμοποιείται η εντολή `Serial.write()` που στέλνει την τιμή της τάσης στη σειριακή που παραλαμβάνει η `Serial.read()`

### Αλγόριθμος

1. Αποθηκεύστε την τιμή του ποτενσιόμετρου στη μεταβλητή `Sensor_Value`
2. Μετατροπή της τιμής που δίνει το ποτενσιόμετρο με αναλογία 0 έως 255.
3. Αποτυπώστε στην οθόνη και στείλτε την τιμή στη σειριακή θύρα.

### Εντολές

1. Διαβάζουμε την αναλογική τιμή απ' το μεσαίο pin του ποτενσιόμετρου

-> **`sensorValue=analogRead(potPin)`**

2. Κάνουμε αντιστοίχιση αναλογικών τιμών (0-1023) σε PWM (0-255)

-> **`Voltage = map(value, 0, 1023, 0, 255);`**

1. για να αποσταλούν τα ψηφιακά δεδομένα σε σειριακή θύρα -συγκεκριμένα η τιμή της τάσης που διαβάζεται από το ποτενσιόμετρο - ώστε μετά να διαβαστεί από το πρόγραμμα σε γλώσσα processing που δημιουργεί τα γραφικά και στην LCD οθόνη.

-> **`Serial.write(Voltage)`**

2. Για την απεικόνιση των δεδομένων στην οθόνη LCD χρειάζεται η βιβλιοθήκη `LiquidCrystal_I2C.h` που περιέχει τις εντολές:

- **`lcd.init();`** : Για την αρχικοποίηση της οθόνης
- **`lcd.backlight();`** : Για να γίνει η οθόνη φωτεινή
- **`lcd.clear();`** : Καθαρίζει την οθόνη και τοποθετεί το δρομέα στην πάνω αριστερή γωνία
- **`lcd.setCursor(15,0);`** : Ο δρομέας της οθόνης τοποθετείται στη στήλη 15 και τη γραμμή 1
- **`lcd.scrollDisplayLeft();`** : Τα περιεχόμενα κυλάνε στην οθόνη κατά μια θέση προς τα αριστερά
- **`lcd.print(array1[positionCounter]);`** : Τυπώνει ένα μήνυμα στην οθόνη ξεκινώντας από τη θέση που βρίσκεται ο δρομέας. Για να γίνει αυτό δημιουργήθηκε ένας πίνακας με ονομασία `array1`.

- `lcd.print("Voltage=")`; : Τυπώνει ένα σταθερό μήνυμα στην οθόνη

4. Με τη μέθοδο `serial.read()` το πρόγραμμα processing επικοινωνεί με το Arduino και με την `serial.write()` μεταφέρει την τιμή της τάσης στο πρόγραμμα.

#### Πρόγραμμα

```

1. #include <LiquidCrystal_I2C.h>
2. #include <Wire.h>
3. int potPin = 0;
4. char array1[]="Counting the Voltage"; //αυτό που φαίνεται στην LCD
5. LiquidCrystal_I2C lcd(0x27,16,2); // ορίζει το αντικείμενο LCD οθόνη για 16
 χαρακτήρες και 2 γραμμές
6. void setup()
7. {
8. lcd.init(); // αρχικοποίηση της οθόνης
9. Serial.begin(9600);
10. lcd.backlight();
11. lcd.clear(); //καθαρίζει την οθόνη από προηγούμενο περιεχόμενο
12. lcd.setCursor(0,0); // τοποθετεί το δρομέα της οθόνης στη στήλη 0, γραμμή 0
13. for (int positionCounter1 = 0; positionCounter1 < 16; positionCounter1++)
14. {
15. lcd.print(array1[positionCounter1]); // τυπώνει το μήνυμα στην LCD οθόνη.
16. delay(1000);
17. }
18. }
19. void loop() {
20. lcd.clear();
21. int sensorValue = analogRead(potPin);
22. int Voltage = map(sensorValue,0,1023,0,255); //επειδή η εντολή serial.read()
 αποδίδει τιμές μεταξύ του 0 και του 255 πρέπει να αλλάξει κλίμακα με τιμές από
 0 έως 1023
23. Serial.write(Voltage);
24. lcd.print("Voltage = ");
25. lcd.print(Voltage); //τυπώνει την πληροφορία στην LCD οθόνη
26. Serial.print("Voltage = ");
27. Serial.println(Voltage); //τυπώνει την πληροφορία στη σειριακή οθόνη
28. delay(200);
29. lcd.clear();
30. }

```

## Εντολές Processing:

### 1. size()

Ορίζει τη διάσταση σε pixels πάνω στο παράθυρο προβολής. Αυτή η συνάρτηση πρέπει να είναι στην πρώτη γραμμή του κώδικα ή η πρώτη μέσα στο setup().

Όποιο μέρος του κώδικα εμφανίζεται πριν τη εντολή size() μπορεί να τρέχει στον κώδικα περισσότερο από μια φορά και αυτό να οδηγήσει σε λάθος αποτελέσματα.

Οι μεταβλητές συστήματος width και height ορίζονται το μέγεθος του παραθύρου προβολής. Αν η size() δεν χρησιμοποιείται, τότε το παράθυρο έχει μέγεθος 100x100 pixels.

### 2. background()

Η συνάρτηση background() ορίζει το χρώμα που χρησιμοποιείται για το φόντο του παραθύρου του Processing. Εξ ορισμού το χρώμα είναι γκρι. Η συνάρτηση χρησιμοποιείται μαζί με τη συνάρτηση draw() για να καθαρίζει ότι υπάρχει μέσα στο παράθυρο αλλά μπορεί και να χρησιμοποιείται μέσα στην setup() ώστε να ορίσει το φόντο.

### 3. fill()

Ορίζει το χρώμα που γεμίζει τα σχήματα.

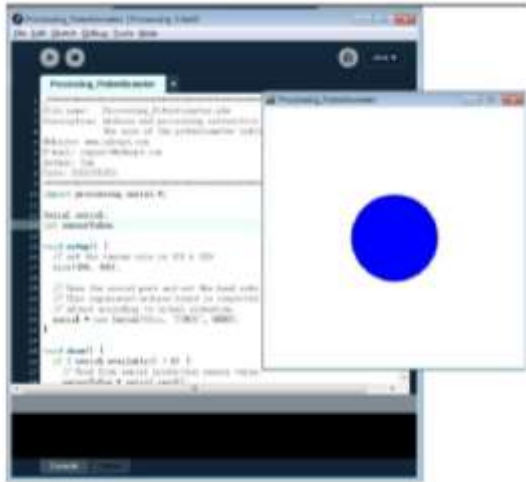
### 4. ellipse()

Σχεδιάζει μια έλλειψη στην οθόνη.


## Πρόγραμμα Processing

Για τη δημιουργία του σχεδίου χρησιμοποιήθηκε η [γλώσσα προγραμματισμού ανοικτού κώδικα Processing](#) που διαθέτει προγραμματιστικό περιβάλλον για προγραμματισμό εικόνων, animation και ήχων. Η γλώσσα αναπτύχθηκε στη Java, το συντακτικό της είναι απλουστευμένο και το προγραμματιστικό της μοντέλο βασίζεται στα γραφικά.

```
1. import processing.serial.*;
2. Serial serial;
3. int sensorValue;
4. void setup() {
5. // ορίζει το μέγεθος του παραθύρου σε 400x400 pixels
6. size(400, 400);
7. // Ανοίγει η σειριακή θύρα με συχνότητα μετάδοσης 9600
8. // Εδώ χρησιμοποιούμε τη θύρα COM5
9. //Προσαρμόστε τις δικές σας συνθήκες
10. serial = new Serial(this, "COM5", 9600);
11. }
12. void draw() {
13. if (serial.available() > 0) {
14. sensorValue = serial.read();
15. //Σχεδιάζει τον κύκλο
16. background(255); // ορίζεται χρώμα φόντου άσπρο
17. fill(0,0,255); // Ορίζεται το χρώμα του κύκλου μπλε
18. ellipse(200, 200, sensorValue, sensorValue); //Το σημείο (200,200) ορίζεται σαν
 το κέντρο του κύκλου
19. }
20. }
21. }
```



### Οδηγίες δοκιμής

1. Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.
2. Στον φάκελό σας θα βρείτε το αρχείο current.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγωγτίστε το και ανεβάστε το στην πλακέτα Arduino.
3. Στο φάκελό σας, επίσης, υπάρχει το αρχείο cycle.pde. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον της γλώσσας προγραμματισμού Processing και βλέπετε το περιεχόμενό του. Πατήστε το πλήκτρο  για να εκτελέσετε το πρόγραμμα και δείτε στο νέο παράθυρο που ανοίγει τον κύκλο που εμφανίζεται.
4. Χρησιμοποιείστε το μοχλό του ποτενσιόμετρου για να αυξομειωθεί η τάση του ρεύματος (δεξιά για αύξηση και αριστερά για μείωση). Παρατηρείστε την αλλαγή στο μέγεθος του κύκλου που είναι ανάλογη με την τάση.

## Ερωτήσεις

1. Περιγράψτε τι κάνουν τα προγράμματα:

.....  
.....

2. Σβήστε το ερωτηματικό «;» από την εντολή `Icd.scrollDisplayLeft();` μεταγλωττίστε και εκτελέστε το πρόγραμμα. Τι παρατηρείτε; Ποιο μήνυμα εμφανίζεται;

.....

2. Τι είδους λάθος δημιουργήσατε;

.....

3. Διορθώστε το λάθος σας και επανεκτελέστε το πρόγραμμά σας.

4. Σε ποιόν ακροδέκτη είναι συνδεδεμένο το μεσαίο pin του ποτενσιόμετρου;

.....

5. Ποια εντολή του προγράμματος Processing χρησιμοποιείται ώστε το εσωτερικό του κύκλου να είναι μπλε;

.....

6. Ποια είναι η εντολή του προγράμματος Processing που ρυθμίζει το μέγεθος του κύκλου;

.....

7. Ποια είναι η χρήση της εντολής `Serial.write(Voltage);`

.....

8. Ποια είναι η χρήση της εντολής `sensorValue = serial.read();`

.....

9. Ποια η χρήση της εντολής: `Icd.clear();`

.....



## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7 - Κατασκευή ραντάρ με Arduino

### Βαθμός δυσκολίας: Δύσκολο

Σε αυτό το μάθημα θα δημιουργήσουμε ένα ραντάρ που με τη βοήθεια ενός αισθητήρα υπερήχων, ένα RGB LED και ένα ηχείο θα μας ειδοποιεί όταν ένα εμπόδιο πλησιάζει προς αυτό. Το RGB LED αλλάζει το χρώμα του όπως ο αισθητήρας υπερήχων ανιχνεύει διαφορετικά αποτελέσματα απόστασης και το ηχείο παράγει ήχους σε διαφορετικές συχνότητες.

### Υλικά

- Arduino UNO Microcontrolier
- USB καλώδιο
- RGB LED
- αισθητήρας υπερήχων
- passive buzzer (ηχείο)
- 1 αντίσταση 1KΩ
- 1 NPN Transistor (S8050)
- 3 αντιστάσεις 220Ω
- 1 Breadboard
- Διάφορα καλώδια σύνδεσης ηλεκτρονικών στοιχείων στο Breadboard

### Περιγραφή κυκλώματος

Χρησιμοποιούμε τον αισθητήρα Υπερήχων HC-SR04 για τον υπολογισμό απόστασης. Η απόσταση που μπορεί να υπολογίσει είναι από 2εκ. έως 400εκ. με ακρίβεια ενός εκατοστού σύμφωνα με τον κατασκευαστή του.



Ο αισθητήρας έχει 4 pins:

- Ground,
- VCC,

- Trig και
- Echo

Το Ground και το VCC pins πρέπει να είναι συνδεδεμένα στο Ground και στο 5 volt pin του Arduino αντίστοιχα. Τα trig και echo pins συνδέονται σε οποιοδήποτε Digital I/O pin του Arduino.

Εκπέμπει έναν υπέρηχο στα 40.000 Hz το οποίο διαδίδεται μέσω του αέρα και αν βρει εμπόδιο από κάποιο αντικείμενο στη διαδρομή του ανακλάται πίσω. Λαμβάνοντας υπόψη το χρόνο της διαδρομής και την ταχύτητα του ήχου μπορείτε να υπολογιστεί η απόσταση.

Για τη λειτουργία του αισθητήρα πρέπει να τεθεί το Trig στο High για 10  $\mu$ s.

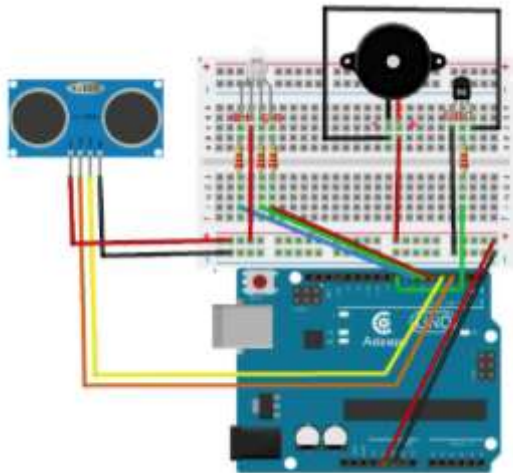
Χρησιμοποιείται η μέθοδος **pulseIn(echoPin, HIGH)** η οποία μετράει το χρόνο σε microseconds μέχρι το echoPin γίνει HIGH και το αποθηκεύει στη μεταβλητή duration.

Για παράδειγμα, αν το αντικείμενο είναι 1 εκατοστό μακριά από τον αισθητήρα και δεδομένου ότι η ταχύτητα του ήχου είναι 340 m/s ή 0.034 cm/ $\mu$ s το ηχητικό κύμα πρέπει να μετακινηθεί περίπου για 29,4  $\mu$ s.

$$\text{Χρόνος}(t) = \text{διάστημα}(s) / \text{ταχύτητα}(v) = 1\text{cm} / 0,034 \text{ cm}/\mu\text{s} = 29,4 \mu\text{s}$$

Αυτή η τιμή που λαμβάνεται από το Echo pin διπλασιάζεται, γιατί το ηχητικό κύμα μετακινείται μπροστά και στη συνέχεια ανακλάται πίσω, άρα η απόσταση που διανύει το κύμα είναι διπλάσια.

Γι αυτό η διάρκεια που παίρνουμε από το Echo pin του αισθητήρα πρέπει να διαιρεθεί με 58,8 (distance=(duration/2)/29,4;) για να υπολογιστεί η απόσταση.



Υλοποίηση πειραματικής διάταξης με Arduino

### Αλγόριθμος

1. Θέσε το trig στο high 10  $\mu$ s
2. Μέτρησε την απόσταση από το εμπόδιο
3. Μετάτρεψέ την απόσταση σε εκατοστά του μέτρου
4. Αν η απόσταση είναι μικρότερη από 20 εκατοστά, το led γίνεται κόκκινο και το ηχείο εκπέμπει ήχο στα 900 Hz  
Διαφορετικά, αν η απόσταση είναι μικρότερη από 120 εκατοστά, το led γίνεται πράσινο και το ηχείο εκπέμπει ήχο στα 300 Hz  
διαφορετικά το led γίνεται κίτρινο και το ηχείο εκπέμπει ήχο διακεκομμένα.

### Συναρτήσεις για το χρωματισμό του LED και της συχνότητας του ηχείου

1. Για τη ρύθμιση των χρωμάτων του RGB LED χρησιμοποιείται η συνάρτηση **setColor** με ορίσματα 3 ακέραιους αριθμούς που παίρνουν τιμές από 0 έως 255. Όταν ο πρώτος παίρνει την τιμή 255 και οι άλλοι δύο το 0, τότε το LED γίνεται κόκκινο. Όταν το μεσαίο παίρνει τιμή 255 και τα άλλα δύο 0, το LED γίνεται πράσινο και όταν οι δύο πρώτοι είναι 0 και ο τρίτος 255 το LED είναι κίτρινο. Με διαφορετικούς συνδυασμούς μπορούμε να επιτύχουμε διαφορετικούς χρωματισμούς.
2. Για τη ρύθμιση της συχνότητας του ηχείου χρησιμοποιούνται οι συναρτήσεις **buzzer1()**, **buzzer2()** και **buzzer3()**. Σε αυτές τις συναρτήσεις χρησιμοποιείται η εντολή `tone(buzzerPin, <συχνότητα>)`. Το `buzzerPin` αντιπροσωπεύει το pin στο οποίο είναι συνδεδεμένο το ηχείο. Για παράδειγμα αν θέλουμε το ηχείο να ηχήσει στα 600 Hz η εντολή πρέπει να είναι: `tone(buzzerPin, 600);`

## Πρόγραμμα

```
1. const int TringPin=3;
2. const int EchoPin=4;
3. int ledPin1 = 5;
4. int ledPin2 = 6;
5. int ledPin3 = 7;
6. int i = 0;
7. int beepPin = 8;
8. void setup()
9. {
10. Serial.begin(9600);
11. pinMode(EchoPin,INPUT);
12. pinMode(TringPin,OUTPUT);
13. pinMode(ledPin1,OUTPUT);
14. pinMode(ledPin2,OUTPUT);
15. pinMode(ledPin3,OUTPUT);
16. pinMode(beepPin,OUTPUT);
17. }
18. void loop()
19. {
20. digitalWrite(TringPin,LOW);
21. delayMicroseconds(2);
22. digitalWrite(TringPin,HIGH);
23. delayMicroseconds(10);
24. int distance = pulseIn(EchoPin,HIGH);
25. distance = distance/58;
26. Serial.println(distance);
27. if (distance<20)
28. {
29. setColor(255,0,0);
30. buzzer1();
31. }
32. else if(distance<=120)
33. {
34. setColor(0,0,255);
35. buzzer2();
36. }
37. else
38. {
39. setColor(0,255,0);
40. buzzer3();
```

```
41. }
42. }
43. void buzzer1()
44. {
45. tone (beepPin,900);
46. delay(300);
47. noTone(beepPin);
48. }
49. void buzzer2()
50. {
51. tone (beepPin,600);
52. delay(300);
53. noTone(beepPin);
54. }
55. void buzzer3()
56. {
57. if (i%100 == 1){
58. tone(beepPin,300);
59. }
60. else{
61. noTone(beepPin);
62. }
63. i++;
64. }
65. void setColor(int red,int green,int blue)
66. {
67. analogWrite(ledPin1,red);
68. analogWrite(ledPin2,green);
69. analogWrite(ledPin3,blue);
70. }
```

### **Οδηγίες**

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο radar.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενο του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

## Ερωτήσεις

1. Περιγράψτε τι κάνει το πρόγραμμα:

.....

.....

.....

2. Σβήστε το ερωτηματικό «;» από την εντολή `digitalWrite(TringPin, HIGH);` μεταγλωττίστε και εκτελέστε το πρόγραμμα. Τι παρατηρείτε; Ποιο μήνυμα εμφανίζεται;

.....

.....

3. Πως ονομάζεται το λάθος που μόλις δημιουργήσατε;

.....

.....

4. Διορθώστε το λάθος σας και επανεκτελέστε το πρόγραμμά σας.

5. Σε ποιούς ακροδέκτες είναι συνδεδεμένο το RGB LED; Που φαίνεται αυτό στο πρόγραμμα;

.....

.....

6. Ποιες εντολές του προγράμματος χρησιμοποιείται για να ανάψει το RGB LED σε χρώμα πράσινο;

.....

.....

7. Πόση πρέπει να είναι η απόσταση του εμποδίου από τον αισθητήρα ώστε το χρώμα του RGB LED να γίνεται πράσινο;

.....

8. Ποια είναι η χρήση της εντολής `int distance = pulseIn(EchoPin,HIGH);`

.....

9. Αλλάξτε το καλώδιο που συνδέει το ηχείο από το pin 8 σε 9. Αλλάξτε το πρόγραμμά σας ώστε το ηχείο να λειτουργεί.

10. Ποια η χρήση της εντολής: `tone(beerPin,300);`

.....

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8 - Πρόγραμμα Που Κινεί Έναν Κινητήρα DC

**Βαθμός Δυσκολίας:** Δύσκολο

Σε αυτό το πείραμα ελέγχεται η κατάσταση του κινητήρα με τη βοήθεια μιας πλατφόρμας Arduino. Οι καταστάσεις που μπορεί να βρίσκεται ο κινητήρας είναι: κίνηση, σταμάτημα, αντίθετη κίνηση, επιτάχυνση και επιβράδυνση.

### Περιγραφή κυκλώματος

Ο DC κινητήρας είναι μια συσκευή που μετατρέπει την ηλεκτρική ενέργεια σε μηχανική. Επειδή εύκολα ελέγχεται, μπορεί να χρησιμοποιηθεί σαν ανεμιστήρας, ηλεκτρονικό παιχνίδι κ.λπ.

Ο κινητήρας DC δουλεύει με συνεχές ρεύμα. Για να συνδεθεί με το Arduino χρειάζονται τέσσερα καλώδια, ένα στη θύρα των 5V, ένα στη γείωση και τα άλλα δύο συνδέονται σε δύο pins (στο δικό μας παράδειγμα είναι το 9 και το 10). Ανάλογα με τη σύνδεση η κίνηση γίνεται δεξιόστροφα ή αριστερόστροφα.

### Υλικά

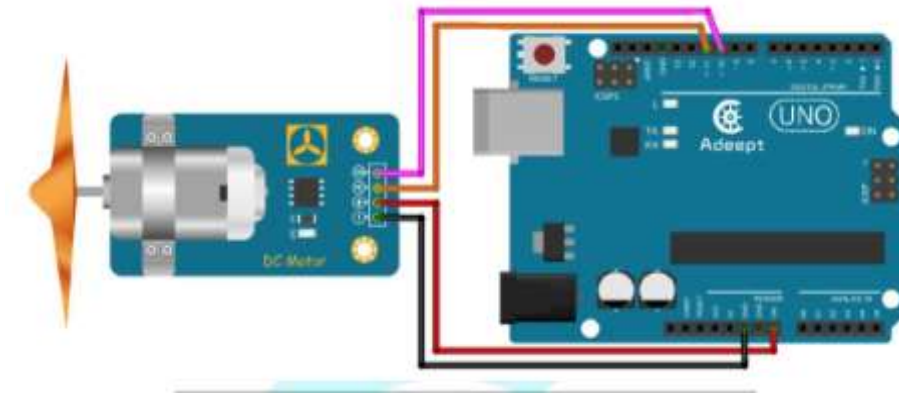
- 1 Arduino UNO R3 Board
- 1 DC Motor Module
- 4 jumper καλώδια

Σημείωση: Πρέπει πρώτα να συνδέουμε πρώτα την τροφοδοσία. Αφού δώσουμε ενέργεια στη συσκευή, συνδέουμε τα pins σήματος. Διαφορετικά ο κινητήρας μπορεί να ζεσταθεί και να μη δουλεύει.

### Περιγραφή του κυκλώματος- Σύνδεση με την πλακέτα

- Το καλώδιο A συνδέεται στο pin 9
- Το καλώδιο B συνδέεται στο pin 10
- Το καλώδιο - συνδέεται στο GROUND
- Το καλώδιο + συνδέεται στο 5V





Όταν φορτωθεί το πρόγραμμα στο Arduino ο ανεμιστήρας, ανάλογα με τις εντολές, γυρίζει, επιταχύνεται, επιβραδύνει, σταματά, ξεκινά κ.ο.κ. Την τιμή της ταχύτητας μπορείτε να τη βλέπετε στη σειριακή οθόνη.

### ΑΣΚΗΣΗ ΠΡΩΤΗ

Σε αυτήν την άσκηση ο κινητήρας κινείται προς τα δεξιά για 5 δευτερόλεπτα, μετά από δύο δευτερόλεπτα κινείται αριστερά και περιμένει δύο δευτερόλεπτα.

### Αλγόριθμος

1. Κινείστε τον κινητήρα δεξιά στη μέγιστη ταχύτητα για 5 δευτερόλεπτα
2. Περίμενε 2 δευτερόλεπτα
3. Κινείστε τον κινητήρα αριστερά στη μέγιστη ταχύτητα για 5 δευτερόλεπτα
4. Περίμενε 2 δευτερόλεπτα
5. Σταμάτησέ τα όλα για 2 δευτερόλεπτα
6. Επανάλαβε τα βήματα 1 έως 5

### Πρόγραμμα

1. `int INA = 9;`
2. `int INB = 10;`
3. `void setup() {`
4. `pinMode(INA,OUTPUT);`
5. `pinMode(INB,OUTPUT);`        `}`
6. `void loop() {`
7. `digitalWrite(INA,255);`
8. `digitalWrite(INB,LOW);`
9. `delay(5000);`

```
10. digitalWrite(INA,LOW);
11. digitalWrite(INB,LOW);
12. delay(200);
13. digitalWrite(INA,LOW);
14. digitalWrite(INB,255);
15. delay(5000);
16. digitalWrite(INA,LOW);
17. digitalWrite(INB,LOW);
18. delay(200); }
```

## Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο dc\_motor1.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενο του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

## ΑΣΚΗΣΗ ΔΕΥΤΕΡΗ

Σε αυτήν την άσκηση ο κινητήρας αυξάνει σταδιακά την ταχύτητά του και μετά από ένα δευτερόλεπτο την μειώνει επίσης σταδιακά.

### Αλγόριθμος

1. Σταμάτησε τον κινητήρα
2. Όρισε την ταχύτητα στο 60
3. Αύξησε την ταχύτητα του κινητήρα κατά 20
4. Περίμενε 1 δευτερόλεπτο
5. Επανάλαβε 7 φορές τα βήματα 3 και 4
6. Μείωσε την ταχύτητα του κινητήρα κατά 20
7. Περίμενε 1 δευτερόλεπτο
8. Επανάλαβε 6 φορές τα βήματα 5 και 6

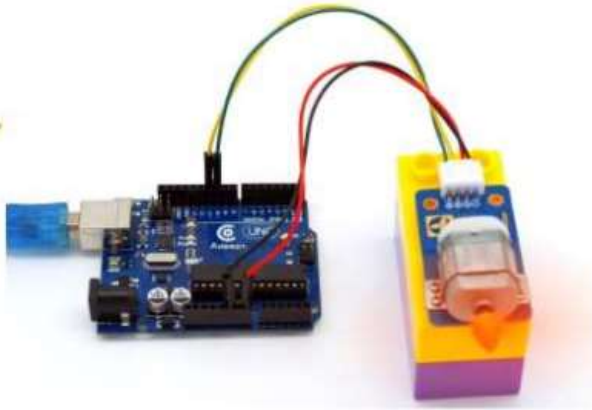
### Εντολές

**digitalWrite(Apin, speed):** η εντολή αυτή κινεί τον κινητήρα με ταχύτητα ίση με την τιμή της μεταβλητής speed προς τα δεξιά.

**digitalWrite(Bpin, speed):** η εντολή αυτή κινεί τον κινητήρα με ταχύτητα ίση με την τιμή της μεταβλητής speed προς τα αριστερά.

## Πρόγραμμα

```
1. void setup()
2. {
3. pinMode(Apin,OUTPUT);
4. pinMode(Bpin,OUTPUT);
5. Serial.begin(9600); //ανοίγει η σειριακή θύρα με ρυθμό στα 9600 bps
6. }
7. void loop()
8. {
9. digitalWrite(Apin,0);
10. delay(1000);
11. DCmotorspeed = 60;
12. for(int i=0;i<7;i++){
13. DCmotorspeed = DCmotorspeed + 20;
14. digitalWrite(Apin,DCmotorspeed);
15. digitalWrite(Bpin,0);
16. Serial.println(DCmotorspeed); //στέλνει δεδομένα στη σειριακή οθόνη
17. delay(1000);
18. }
19. for(int i=0;i<6;i++){
20. DCmotorspeed = DCmotorspeed - 20;
21. digitalWrite(Apin,0);
22. digitalWrite(Bpin,DCmotorspeed);
23. Serial.println(DCmotorspeed); //στέλνει δεδομένα στη σειριακή οθόνη
24. delay(1000);
25. DCmotorspeed = DCmotorspeed - 20;
26. }
27. digitalWrite(Apin,0);
28. digitalWrite(Bpin,0);
29. }
```



### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο `dc_motor2.ino`. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

### Ερωτήσεις

1. Με ποια εντολή κινείται ο κινητήρας DC;

.....

2. Ποιο το αποτέλεσμα της εντολής: `digitalWrite(Apin, 0);`

.....

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9 - Ρύθμιση Έντασης Led Με Αισθητήρα Bluetooth

**Βαθμός Δυσκολίας:** Δύσκολο

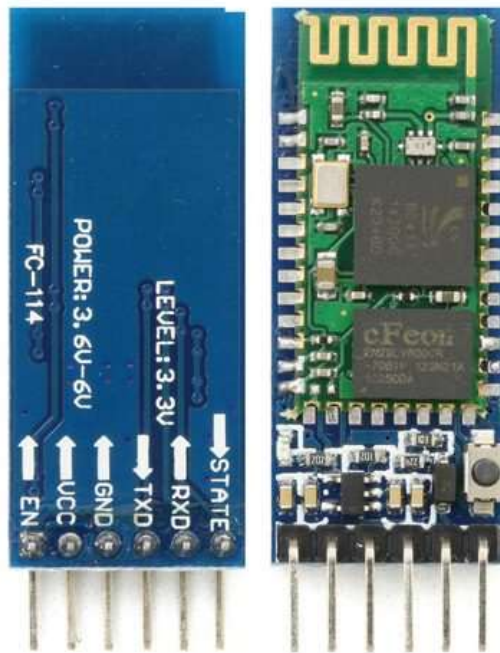
### Περιγραφή

Σε αυτήν την άσκηση θα αναβοσβήνουμε ένα led από smartphone ή tablet χρησιμοποιώντας έναν αισθητήρα Bluetooth συνδεδεμένο με Arduino. Εκτός από τα υλικά του Arduino, θα κατεβάσουμε μια εφαρμογή Android που ονομάζεται BlueControl για να ελεγχθεί το led μέσω αυτής.

**Εφαρμογές:** Με τους αισθητήρες Bluetooth μπορούμε να επικοινωνούμε απομακρυσμένα με άλλες συσκευές όπως smartphones ή tablets και να διαχειριζόμαστε συσκευές όπως οικιακές συσκευές, οχήματα κ.λπ.

Οι δυο πιο δημοφιλείς μονάδες Bluetooth της αγοράς είναι:

- Το **HC-05** με το οποίο παρέχεται η δυνατότητα του να λειτουργεί ως master και slave δηλαδή και στέλνει και δέχεται αιτήσεις και,
- το **HC-06** που διαθέτει 2 παραπάνω pins και λειτουργεί μόνο ως slave (λαμβάνει μόνο αιτήσεις σύνδεσης)



Η μονάδα HC-05 έχει 6 pins:

- VCC , στο οποίο συνδέουμε την τροφοδοσία από το Arduino.
- GND, στο οποίο συνδέουμε την γείωση.
- RX, στο οποίο έρχονται τα δεδομένα από το Arduino.
- TX, από το οποίο αποστέλλονται τα δεδομένα που λαμβάνει η μονάδα Bluetooth προς το Arduino.
- STATE, το οποίο είναι απλά συνδεδεμένο με το LED της μονάδας Bluetooth και όταν αυτό ανάβει βγάζει έξοδο HIGH αλλιώς βγάζει έξοδο LOW.
- ENABLED ή KEY το οποίο μας επιτρέπει να αλλάζουμε την κατάσταση της μονάδας μεταξύ δύο:
  - Κατάσταση δεδομένων, όπου το HC-05 λειτουργεί κανονικά για να μεταφέρει δεδομένα.
  - Κατάσταση ρυθμίσεων, όπου μπορούμε να δίνουμε εντολές στο HC-05 για να αλλάζουμε τις ρυθμίσεις του.

Επίσης διαθέτει μια ένδειξη LED η οποία ανάλογα με τον ρυθμό που αναβοσβήνει μας πληροφορεί για την κατάσταση της μονάδας, καθώς και ένα πλήκτρο το οποίο μας επιτρέπει να αλλάζουμε την κατάσταση της μονάδας.

### **Σύνδεση με Arduino**

Για να συνδεθεί το HC-05 με το Arduino χρειάζεται να στέλνει και να λαμβάνει δεδομένα αξιοποιώντας τα pins 0 και 1 του Arduino τα οποία έχουν την ειδική σήμανση RX και TX αντίστοιχα. Το pin 0 (RX) του Arduino δέχεται (Receive) δεδομένα από την μονάδα Bluetooth, ενώ το pin 1 (TX) του Arduino στέλνει (Transmit) δεδομένα στην μονάδα Bluetooth. Η σύνδεση λοιπόν που πρέπει να κάνουμε είναι η εξής:

HC-05 VCC → Arduino 5V

HC-05 GND → Arduino GND

HC-05 TX → Arduino RX (pin 0)

HC-05 RX → Arduino TX (pin 1)

### **Μεταφέροντας το πρόγραμμα στο Arduino**

Μπορούμε πλέον να γράφουμε προγράμματα στο Arduino τα οποία να στέλνουν και να δέχονται δεδομένα μέσω Bluetooth χρησιμοποιώντας τις εντολές της σειριακής

επικοινωνίας που περιλαμβάνονται στην βιβλιοθήκη Serial. Οι ίδιες εντολές χρησιμοποιούνται και για την επικοινωνία μέσω USB (και αυτή σειριακή είναι).

**Σημείωση:** Όταν θέλουμε να ανεβάσουμε το πρόγραμμα στο Arduino από τον υπολογιστή μας χρησιμοποιώντας την θύρα USB θα πρέπει να προσέξουμε να απενεργοποιήσουμε προσωρινά την μονάδα Bluetooth (απλά αποσυνδέστε το καλώδιο στο pin VCC, ή τα καλώδια στο RX και TX pins) για να μπορέσει να ανέβει το πρόγραμμα μας. Αν ξεχάσουμε να αποσυνδέσουμε το Bluetooth, το Arduino IDE θα μας ειδοποιήσει ότι η σειριακή θύρα του Arduino είναι απασχολημένη (αφού την έχει δεσμεύσει το Bluetooth) και δεν θα ανεβάσει το πρόγραμμα. Αφού ανεβάσουμε το πρόγραμμα μας στο Arduino μπορούμε να συνδέσουμε και πάλι την μονάδα HC-05 ώστε να λειτουργήσει κανονικά.

### Υλικά

- Arduino
- Bluetooth συσκευή/αισθητήρας – HC05
- Καλώδια
- LEDs
- Μία συσκευή ANDROID

### Η διαδικασία

Κατεβάστε την εφαρμογή Android που ονομάζεται **BlueControl** (δωρεάν). Αυτόματα ενεργοποιεί το Bluetooth της συσκευής σας.

- Συνδέστε τον αισθητήρα όπως φαίνεται στην πιο πάνω εικόνα. Μη συνδέσετε τα RX & TX pins πριν ανεβάσετε τον κώδικα!
- Ανεβάστε στο Arduino τον πιο πάνω κώδικα.
- Την πρώτη φορά που η συσκευή σας δει το Bluetooth αισθητήρα, σας ζητάει password που είναι ή 1234 ή 0000.
- Όταν η συσκευή συνδεθεί με τον αισθητήρα, το LED θα ξεκινήσει να αναβοσβήνει με πιο αργό ρυθμό από το συνηθισμένο.

## Αλγόριθμος

1. Διάβασε τα δεδομένα από τον αισθητήρα Bluetooth
2. Τύπωσε τα δεδομένα στη σειριακή θύρα
3. Αν το σήμα είναι 1 (δηλαδή έχει πατηθεί ο χαρακτήρας A στην εφαρμογή BlueControl)  
Άναψε το led
4. Αν το σήμα είναι 0 (δηλαδή έχει πατηθεί ο χαρακτήρας B στην εφαρμογή BlueControl)  
Σβήσε το led

## Εντολές

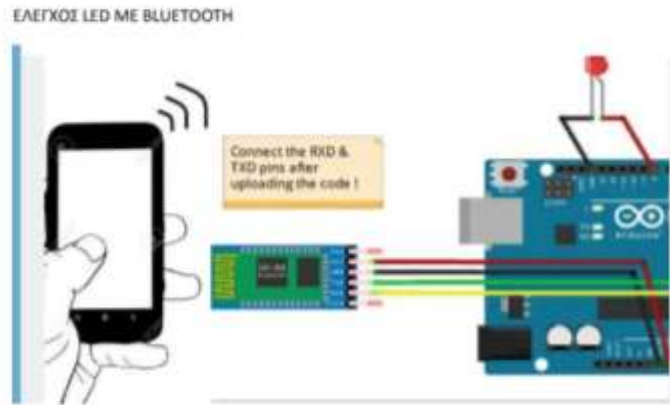
Χρησιμοποιείται η εντολή switch που ανήκει στην κατηγορία των εντολών επιλογής όπως και η if.

## Πρόγραμμα

```
void setup() {
 Serial.begin(9600);
 pinMode(13, OUTPUT);
}
void loop() {
 if(Serial.available()>0) {
 char data= Serial.read();
 Serial.print(data);
 switch(data) {
 case '1': digitalWrite(13, HIGH);break; // όταν πατηθεί το a στην εφαρμογή
BlueControl το led ανάβει
 case '0': digitalWrite(13, LOW);break; // όταν πατηθεί το d στην εφαρμογή
BlueControl το led σβήνει
```



```
default : break; } // στις υπόλοιπες περιπτώσεις δεν συμβαίνει κάτι
 Serial.println(data); }
delay(50); }
```



## Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο blh.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

## Παράρτημα

### Διαχείριση του αισθητήρα Bluetooth από τη σειριακή θύρα του Arduino

Μόλις ανοίξουμε την σειριακή οθόνη θα εμφανιστούν αρχικά οι πληροφορίες που έχουμε βάλει στην συνάρτηση setup του προγράμματος.

### Έλεγχος λειτουργίας

Για να ελέγξουμε αν όλα έχουν συνδεθεί σωστά και μπορεί το HC-05 να δεχτεί εντολές μπορούμε να στείλουμε την εντολή AT, η οποία θα πρέπει να μας επιστρέψει OK που σημαίνει ότι η σύνδεση είναι σωστή.

Αν όλα πάνε καλά η εντολή AT θα μας επιστρέψει OK.

### Διαχείριση ονόματος της συσκευής

Για να δούμε το όνομα που έχει ήδη η μονάδα HC-05 πληκτρολογούμε την εντολή AT+NAME.

Το όνομα της μονάδας είναι αυτή τη στιγμή HC-05

Για να αλλάξουμε το όνομα γράφουμε την εντολή AT+NAME=Νέο όνομα. Έτσι, αν για παράδειγμα φτιάχνουμε έναν έξυπνο φωτισμό όπου θα συνδέεται με το κινητό μας μέσω Bluetooth για να στέλνει δεδομένα, μπορούμε να δώσουμε την εντολή AT+NAME=Smart Light.

Αλλάζουμε το όνομα του HC-05 σε Smart Light.

Για να ελέγξουμε αν όντως έγινε η αλλαγή, μπορούμε να δώσουμε ξανά την εντολή AT+NAME για να δούμε το όνομα της μονάδας.

Δίνοντας την εντολή AT+NAME βλέπουμε πως το όνομα του HC-05 έχει πλέον αλλάξει.

Για να αλλάξουμε τον κωδικό της μονάδας όταν γίνεται η σύζευξη με άλλες συσκευές Bluetooth χρησιμοποιούμε την εντολή AT+PSWD=<Νέος κωδικός>

Δίνοντας την εντολή AT+PSWD βλέπουμε τον τρέχοντα κωδικό της μονάδας

Αλλάζουμε τον κωδικό σε 0000 με την εντολή AT+PSWD=0000

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 10 – Πρόσβαση Ασφαλείας Με Αναγνώστη RFID

**Βαθμός Δυσκολίας:** Δύσκολο

### Περιγραφή

Το **RFID** είναι τα αρχικά του όρου: **Radio Frequency Identification** (ταυτοποίηση μέσω ραδιοσυχνοτήτων). Πρόκειται για ένα σύστημα ταυτοποίησης που χρησιμοποιεί ραδιοσυχνότητες για ταυτοποίηση συσκευών με αναγνώριση και ανίχνευση.

Το RFID χρησιμοποιεί ηλεκτρομαγνητικά πεδία για να μεταφέρει δεδομένα σε μικρές αποστάσεις.

Αποτελείται από δύο μέρη:

A) **πομποδέκτης** (transponders) ή RFID tags – ετικέτες που περιλαμβάνει μνήμη ώστε να αποθηκεύει δεδομένα και μια κεραία. Κάθε tag έχει τη δική της ταυτοποίηση (UID).

B) **αναγνώστες** – αισθητήρες (readers) που ανακτούν δεδομένα από τις ετικέτες RFID. Έχουν ενσωματωμένα μια κεραία και μια μονάδα ελέγχου. Είναι δύο κατευθύνσεων ραδιοσυχνότητες που στέλνουν το σήμα στην ετικέτα (tag) και διαβάζουν την απάντηση.

Σε αυτή την άσκηση θα δούμε πως γίνεται η επικοινωνία μεταξύ του αναγνώστη και του πομποδέκτη μέσω της σειριακής οθόνης. Μια μελλοντική επέκταση της άσκησης είναι η κατασκευή ενός κουτιού που η κλειδαριά του θα ανοίγει με τη συσκευή RFID.

### Εφαρμογές:

Μπορεί να χρησιμοποιηθεί για να ανοίξει μια πόρτα. Δηλαδή μόνο το πρόσωπο με τη σωστή πληροφορία στην κάρτα του μπορεί να ανοίξει την πόρτα. Άλλη εφαρμογή είναι στα προϊόντα που έχουν συρμάτινες ταινίες στις αλυσίδες καταστημάτων. Είναι χρήσιμο για την αναγνώριση προσώπων, να γίνουν συναλλαγές κ.λπ.



**Βασικές προδιαγραφές:**

- Τάση εισόδου: 3.3V
- Συχνότητα: 13.56MHz

**Το κύκλωμα**

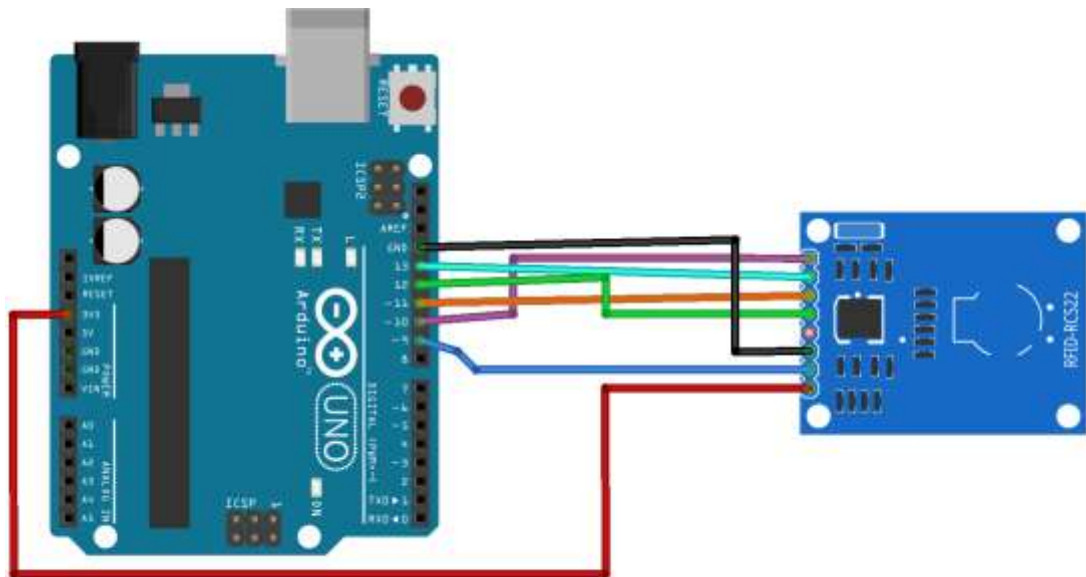
**1. Υλικά**

- Arduino UNO
- SparkFun RFID Reader Breakout
- Adafruit RFID/NFC card

**2. Σύνδεση του Arduino με το RFID reader.**

Βρείτε το PIN της κάρτας και ακολουθείστε τη συνδεσμολογία όπως φαίνεται παρακάτω:

- SDA-----Digital 10
- SCK-----Digital 13
- MOSI-----Digital 11
- MISO-----Digital 12
- IRQ-----unconnected
- GND-----GND
- RST-----Digital 9
- 3.3V-----3.3V (DO NOT CONNECT TO 5V)



fritzing

## Η διαδικασία

### 1. Μεταφέροντας το πρόγραμμα στο Arduino

Για να λειτουργήσει η συσκευή χρειάζεται η βιβλιοθήκη *rfid-master* και να προστεθεί στον φάκελο των βιβλιοθηκών του Arduino (μετά από αυτό πρέπει να γίνει επανεκκίνηση του Arduino IDE ώστε να αναγνωριστεί από το Arduino).

### 2. Αλγόριθμος

- a. Ο αναγνώστης ψάχνει την κάρτα
- b. Όταν η κάρτα πλησιάσει στον αναγνώστη, τυπώνει το UID στη σειριακή οθόνη
- c. Αν το UID που έχει καταχωρηθεί τον αναγνώστη είναι ίδιο με αυτό της κάρτας τότε τυπώνεται το μήνυμα «Εξουσιοδοτημένη πρόσβαση» διαφορετικά αρνείται την πρόσβαση.

### 3. Διάβασμα των δεδομένων από το RFID tag

Αφού ετοιμαστεί το κύκλωμα, στο Arduino IDE επιλέξτε: Αρχείο > Παραδείγματα > MFRC522 > DumpInfo και ανεβάστε τον κώδικα. Αυτός ο κώδικας είναι διαθέσιμος στο Arduino IDE (αφού έχει εγκατασταθεί η βιβλιοθήκη RFID).

Το πρόγραμμα είναι το παρακάτω:

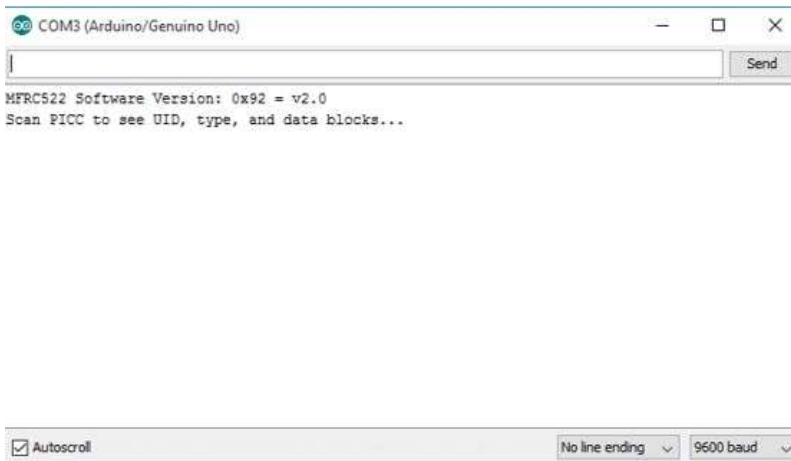
```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Δημιουργείται το στιγμίοτυπο MFRC522.

void setup()
{
 Serial.begin(9600); // Αρχικοποιεί τη σειριακή οθόνη
 SPI.begin(); // Αρχικοποιεί το SPI bus
 mfrc522.PCD_Init(); // Αρχικοποιεί το MFRC522
 Serial.println("Πλησιάστε την κάρτα στον αναγνώστη...");
 Serial.println();
}
}
```

```
void loop()
{
 // Ψάχνει για νέα κάρτα
 if (! mfrc522.PICC_IsNewCardPresent())
 {
 return;
 }
 // Επιλέγει μια κάρτα (αυτή που έχει πλησιάσει)
 if (! mfrc522.PICC_ReadCardSerial())
 {
 return;
 }
 Serial.print("UID tag :"); //Τυπώνει το UID στη σειριακή οθόνη
 String content= "";
 byte letter;
 for (byte i = 0; i < mfrc522.uid.size; i++)
 {
 Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
 Serial.print(mfrc522.uid.uidByte[i], HEX);
 content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
 content.concat(String(mfrc522.uid.uidByte[i], HEX));
 }
 Serial.println();
 Serial.print("Μύνημα : ");
 content.toUpperCase();
 if (content.substring(1) == "BD 31 15 2B") //αλλάξτε εδώ το UID της κάρτας /ών
 που θέλετε να δώσετε πρόσβαση
 {
 Serial.println("Εξουσιοδοτημένη πρόσβαση");
 Serial.println();
 delay(3000);
 }
 else {
 Serial.println(" Δεν επιτρέπεται η πρόσβαση");
 delay(3000);
 }
}
```

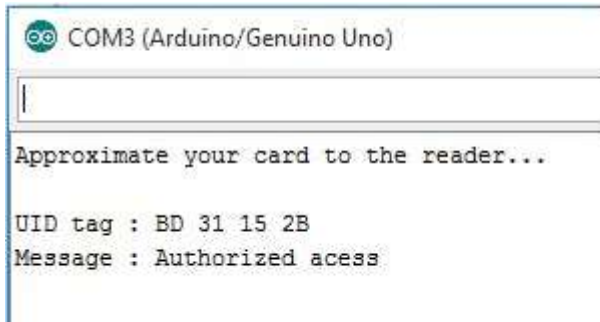
Όταν ανοίξει η σειριακή οθόνη φαίνεται το παρακάτω:



Πλησιάστε την RFID κάρτα ή το μπρελόκ με τον αναγνώστη μέχρι να εμφανιστεί η πληροφορία, όπως φαίνεται στην επόμενη εικόνα:

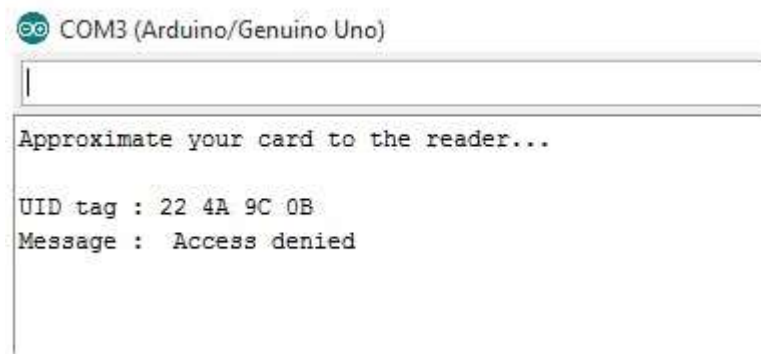






```
COM3 (Arduino/Genuino Uno)
|
Approximate your card to the reader...
UID tag : BD 31 15 2B
Message : Authorized access
```

Αν πλησιάσετε μια άλλη κάρτα με άλλο UID, θα εμφανιστεί το μήνυμα που ενημερώνει ότι δεν αναγνωρίζει το UID:



```
COM3 (Arduino/Genuino Uno)
|
Approximate your card to the reader...
UID tag : 22 4A 9C 0B
Message : Access denied
```

### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο rfid.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

Πλησιάστε την κάρτα που διαλέξατε να δίνει πρόσβαση και τότε θα φανεί το μήνυμα που επιτρέπει ή όχι την πρόσβαση.

## Βιβλιογραφία

### Άρθρα

Brousseau, G. (1984). The crucial role of the didactical contract in the analysis and construction of situations in teaching and learning mathematics. In H.-G. Steiner (Ed.), *Theory of mathematics education: ICME 5 – topic area and miniconference: Adelaide, Australia*. Bielefeld, Germany: Institut fuer Didaktik der Mathematik der Universitaet Bielefeld.

Chionidou-Moskofoglou, M (2005). Βασικές Μέθοδοι Ομαδοσυνεργατικής Διδασκαλίας και Μάθησης στα Μαθηματικά. [http://users.auth.gr/~lemonidi/sde\\_yliko/Omadosinergatik%20didaskalia%20.pdf](http://users.auth.gr/~lemonidi/sde_yliko/Omadosinergatik%20didaskalia%20.pdf)

Gonzalez, H. B., & Kuenzi, J. J. (2012). *Science, technology, engineering, and mathematics (STEM) education: A primer*. Congressional Research Service. Ανακτήθηκε 14 Σεπτεμβρίου 2019 από <http://www.fas.org/sgr/crs/misc/R42642.pdf>.

Kotini I., & Tzelepi S. (2013). Σχεδιάζοντας ένα μάθημα Πληροφορικής που βασίζεται στις αρχές του Εποικοδομισμού, της Υπολογιστικής σκέψης και της παιχνιδοποίησης. Στο 5<sup>ο</sup> συνέδριο «Η πληροφορική στην εκπαίδευση». 11-13/10/2013. Πειραιάς: Πανεπιστήμιο Πειραιώς.

Psycharis, S. (2015). 'The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education ; *Journal of Science Education, and Technology* 25(2),316-326 (JOST) DOI 10.1007/s10956-015-9595-z

Psycharis, S., (2016). 'Inquiry Based- Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education (Accepted for publication at *Journal "Educational Technology & Society"* - Volume 19, Issue 3, 2016.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49, 33-35.

### **Βιβλία**

Γρηγοριάδου, Μ., κ.α., Διδακτικές προσεγγίσεις και εργαλεία για τη διδακτική της Πληροφορικής, Εκδόσεις Νέων Τεχνολογιών, 2009.  
ISBN: 978-960-67-5923-9

Νούσης, Β., (2019), Arduino για Αρχαρίους, Β. Νούσης: Ηγουμενίτσα  
ISBN : 978-618-83502-3-6

Παλιούρας, Α., (2017), Κατασκευάζω και προγραμματίζω με τον μικροελεγκτή Arduino, [http://robotics-edu.gr/data/arduino/arduino\\_paliouras.pdf](http://robotics-edu.gr/data/arduino/arduino_paliouras.pdf), ISBN: 978-960-93-8945-7

Πουλάκης, Ε. (2015). Προγραμματίζοντας με τον μικροελεγκτή Arduino, Ε. Πουλάκης: Ηράκλειο <http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf>  
\_ISBN 978-960-93-6760-8

### **Διαδικτυακοί τόποι**

<http://arduino.cc>

<http://create.arduino.cc/projecthub>

<https://4dimkal-robot.weebly.com/tiota-epsilon943nualphaiota-tauomicron-arduino.html>

[https://el.wikipedia.org/wiki/Εκπαιδευτική\\_ρομποτική](https://el.wikipedia.org/wiki/Εκπαιδευτική_ρομποτική)

<https://opencourses.auth.gr/modules/document/index.php?course=OCRS370&openDir=/5576cca6CIm7>

**ISBN: 978-618-00-3211-6**