ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ ARDUINO:ΣΧΕΔΙΑΣΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ

ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΚΩΝΣΤΑΝΤΙΝΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Επίκουρος Καθηγητής: ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΗΣ.

Λαμία 7/6 έτος 2025

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

THE PROGRAMMING LANGUAGE INTO ARDUINO:DESIGN AND SIMULATION

KONSTANTINOS PAPAKONSTANTINOU

FINAL THESIS

ADVISOR

Assistant professor: DADALIARHS ANTONHS

Lamia 7/6 year 2025

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 7/6/2025

Ο – Η Δήλων.
ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΚΩΝΣΤΑΝΤΝΟΥ

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία σχετίζεται με την μελέτη και την έρευνα του **Arduino**, με παραδείγματα κυκλωμάτων σε φυσική μορφή(**Pins,Inputs,Outputs,Chips)** αλλά ακόμη και την ψηφιακή σχεδίαση σε διάφορες πλατφόρμες όπου μπορούμε να υλοποιήσουμε σχεδιασμό κυκλωμάτων **Arduino**.(**Tinkercad, Woki, SimuLIDE).** Θα μελετήσουμε συγκρίσεις σε διαφορετικές πλακέτες, με θετικά και αρνητικά για το κάθε ένα(**UNO R2, UNO R3)**.Στην συνέχεια, υπάρχει εξήγηση σε διάφορα εξαρτήματα και παραδείγματα συστημάτων **Arduino.** Επιπλέον, τρόπους εμπέδωσης και κατανόησης στις προσομοιώσεις που θα πραγματοποιήσουμε .Επίσης,θα συγκρίνουμε τις πλατφόρμες σχεδίασης,με τα διάφορα χαρακτηριστικά, μετά θα αρχίσουμε με την πρώτη και την δεύτερη σχεδίαση κυκλώματος **Arduino,** βήμα προς βήμα με ανάλυση κώδικα των παραδειγματων μας αλλά, παράλληλα θα δούμε και τα εξαρτήματα που θα χρησιμοποιήσουμε **(LED, BUZZER ,BUTTON, DISTANCE SENCOR)**,με ποιον τροπο θα τα συνδέσουμε και το αποτέλεσμα τους.Τελος, θα αναλύσουμε την γλώσσα **Arduino** και διάφορες συναρτήσεις που υπάρχουν στην γλώσσα προγραμματισμού του **(DIGITALI/O, ANALOG I/O, ADVANCED I/O).**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ABSTRACT

The thesis is about the study and research of **Arduino**,including different examples of physical circuits**(Pins,Inputs,Outputs,Chips)** but also in digital form a variety of digital platforms,where they can illustrate a design or programming of **Arduino** circuits **(Tinkercad, Woki, SimulIDE)**. We will study the differences, benefits and negatives of various motherboards **(Uno R2,Uno R3).** Furthermore, we will include explanation and the usage of different electronic components . Additional emphasis is placed on ways to reinforce understanding through the simulations we will carry out. Moreover,we will compare the design platforms based on their features. Then we will proceed with the first and second **Arduino** circuit design,explained step by step ,including code analysis of each example,At the same time ,we will examine components used**(LED,BUZZER,BUTTON,DISTANCE SENSOR)**,on how they connect and the outcomes they produce. Finally, there will be a chapter based on the **Arduino** programming language with analyses,along with several functions it supports **(DIGITALI/O, ANALOG I/O, ADVANCED I/O).**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΠΕΡΙΕΧΟΜΕΝΑ

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΕΙΣΑΓΩΓΗ

Με την άνοδο της τεχνολογια υπάρχει σημαντική εξέλιξη στον αυτοματισμό και στην ηλεκτρονική, με αυτό πήραν διάφορες μορφές έξυπνων συστημάτων όπου άρχισαν να μας βοηθούν στην καθημερινότητα. Αλλά από τότε που άρχισε να υπάρχει ο προγραμματισμός, έγινε τεράστια αύξηση στο τι μπορούμε να κάνουμε και σε πολλά ηλεκτρικά και υπολογιστικά συστήματα. Αρχίσαμε να βρίσκουμε ευκολίες σε πολλά πράγματα που κάνουμε καθημερινά χρησιμοποιώντας της εφευρέσεις μας και μια από αυτές είναι το **Arduino** .

Το **Arduino[1]** είναι ένα open source **hardware** το οποίο έχει ενσωματωμένους μικροελεγκτές και εισόδους και εξόδους. Φτιάχτηκε το 2005 ως ένα σχέδιο μια συσκευής για τον έλεγχο προγραμμάτων και διαδραστικών σχεδίων από μαθητές επειδή ήταν η ποιό φθηνή από άλλα πρωτότυπα. Πρέπει να σημειωθεί επίσης ότι προγραμματίζεται από την γλωσσά **Wiring** ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού **C++** και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην **C++** .Η πρώτη πλακέτα που τα ξεκίνησε όλα ήταν το **Arduino Uno**,όπου αυτό αποτελεί την βάση για ολες τις άλλες εκδόσεις και έφερε μια επανάσταση στο χώρο του ηλεκτρονικού σχεδιασμού οπου αυτό είχε χαρακτηριστικά:Έναν μικροελεγκτή **ATmega8**, 14 **pins** (Ψηφιακοί Είσοδοι/Έξοδοι), αναλογικούς είσοδους 6.Η τροφοδότηση ηταν μεσω **USB**, η εξωτερικοί τροφοδότες **(7-12V)** και τέλος για να συνδεθούμε με αυτό και να το προγραμματίσουμε είχε μια θύρα **USB type B**, Θα δούμε αναλυτικά στις επόμενες σελίδες που χρησιμεύουν το κάθε ένα χαρακτηριστικό.Υπάρχουν πολλά παρόμοια με τα **Arduino**, όπως ένα **Raspberry Pi** η τα **ESP32** και τα **Weensy Boards**,προφανώς αυτά λίγα παραδείγματα που παρομοιάζουν άλλα ταυτόχρονα δεν είναι ίδια γιατί κάθε ένα από αυτά έχει κάτι ξεχωριστό στην πλακέτα που βρίσκεται. Για παράδειγμα τα **Raspberry** λειτουργούν ποιο πολύ ως ένα μικρό υπολογιστή πάρα μια πλακέτα που μπορούμε εύκολα να ενσωματώσουμε και να δοκιμάσουμε εξαρτήματα και προεκτάσεις. Μπορεί να βάλει ένας λειτουργικό σύστημα(πχ **Linux**).Είναι κατάλληλα για διάφορα πρότζεκτ όπως να φτιάξεις ένα **Server** ή από τα πιο γνωστά πρότζεκτ είναι να το χρησιμοποιείς ως σύστημα παιχνιδιών επειδή η πλακετα έχει ενσωματωμένα συστήματα όπως κάρτα **wifi**, **Bluetooth** και **HDMI port** προφανούς και για αυτό υπάρχει διαφορά τιμής στο πλαίσιο αγοράς. Επίσης η διαφορά του **Arduino** με παράδειγμα ένα **Weensy board** είναι κυρίως ότι το **Arduino** είναι ένα από τα πιο προσβάσιμα συστήματα στην αγορά και είναι αρκετά εύκολο να καταλάβει κάποιος το πως λειτουργεί και πως να κάνει ένα δικό του project σε μικρότερο χρόνο,ενώ το άλλο σύστημα προφανώς είναι και πιο δυνατό,έχει μεγαλύτερη ταχύτητα και μπορεί να επεξεργαστεί δεδομένα (ήχου,βίντεο,κ.α) με ποιο εύκολη δυνατότητα,δεν έχει αυτό το βασικό στοιχείο που έχει ένα **Arduino** οποί έχει μια από τις ποιο εύκολες και προσβάσιμες χρησεις.

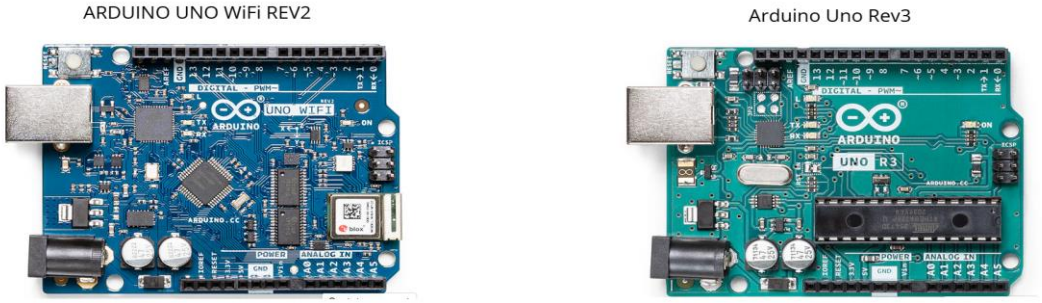
Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Εμείς θα δούμε και τις δωρεάν προσβάσιμες πλατφόρμες άλλα και διάφορα σχεδία έτσι ώστε να κατανοήσουμε την τέχνη του σχεδιασμού και προγραμματισμού ενός συστήματος,με αυτόν τον τρόπο θα μάθουμε πω να κάνουμε **mini projects** και έτσι θα καταλάβουμε τα **"basics"** των ηλεκτρονικών συστηματων. Το **Arduino** είναι ένας από τους πιο γνωστούς λόγους και τρόπους για την κατανόηση, τον αυτοματισμό με προγραμματισμό και διάφορους αισθητήρες σε κάθε περιβάλλον που υπάρχει.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΚΕΦΆΛΑΙΟ 2 Arduino

Αφού είδαμε με το τι θέμα θα ασχοληθούμε τώρα πρέπει να δούμε μερικές βασικές γνώσεις του **Arduino** έτσι,μετά να είναι πιο εύκολα να καταλάβουμε στα παραδείγματα μας .Στα παραδείγματα μας όταν θα μιλάμε για **Arduino** πλακέτα θα χρησιμοποιούμε την πιο γνωστή και γενικά πιο προσβάσιμη και οικονομικά αλλά και σε ότι υπάρχει σε παραδειγματα. Αυτη η πλακέτα είναι το **Arduino** από την οικογένεια **Uno,R3**.Γενικά υπάρχουν πιο καλύτερες από αυτήν και στην ίδια οικογένεια για παραδείγματα όπως την **Uno R2 WiFi** όπου έχει και **Bluetooth** δυνατότητες και **WiFi chip** αλλά αυτές για πρώτη γνώση μπορεί να είναι πιο πολύ τα προσβάσιμα στοιχεία τους και γενικά να υπάρχουν μεγαλύτερες αποδόσεις ταχύτητας,όμως χρειάζεται πρώτα να δείξουμε τρόπους έτσι ώστε ένας άγνωστος στον τομέα αυτό θα καταλάβει ένα και παραπάνω παραδείγματα με εύκολο τροπο. Θα δούμε την διαφορά τους και μετά θα συνεχίσουμε με την αναπαράσταση την πλακέτας που έχουμε

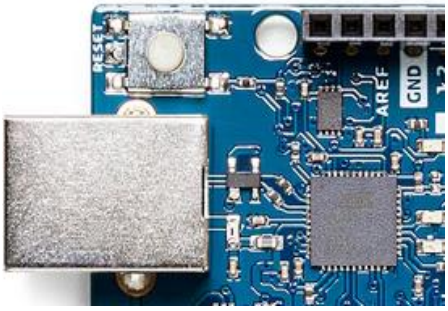


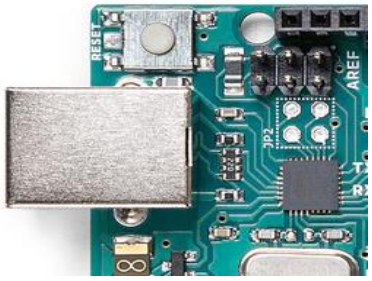
Εικονα 2.0(Arduino Uno Wifi Rev2 & Arduino Uno Rev3)

Καταρχήν,οι διαφορές είναι πάρα πολύ προφανές και με το πρωτοβλέπουμε , αλλά και μετά όπως θα αναλύσουμε την πιο προγνωστική πλακέτα,με τα χαρακτηριστικά τους και στον τομέα των εξαρτημάτων,οπότε ας ξεκινήσουμε με την σύγκριση:

Πρώτον, πέρα από τις πιο προφανές διαφορές ονόματος,όπου το ένα αναφέρεται ως **Uno WiFi** και το άλλο **Uno R3**.Βλέπουμε ότι ο δεύτερος μικροελεγκτής της θύρας στο **Rev2** είναι πιο ενισχυμένος,για απευθείας επικοινωνία με τον επεξεργαστή,επειδή έχει διαφορετικό chip :

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156





Επίσης,ένα σημαντικό διαφορετικό σημείο έχει να κάνει με το **WiFi chip** όπου αναφέρεται ως **u-blox**

πάνω σε αυτο:

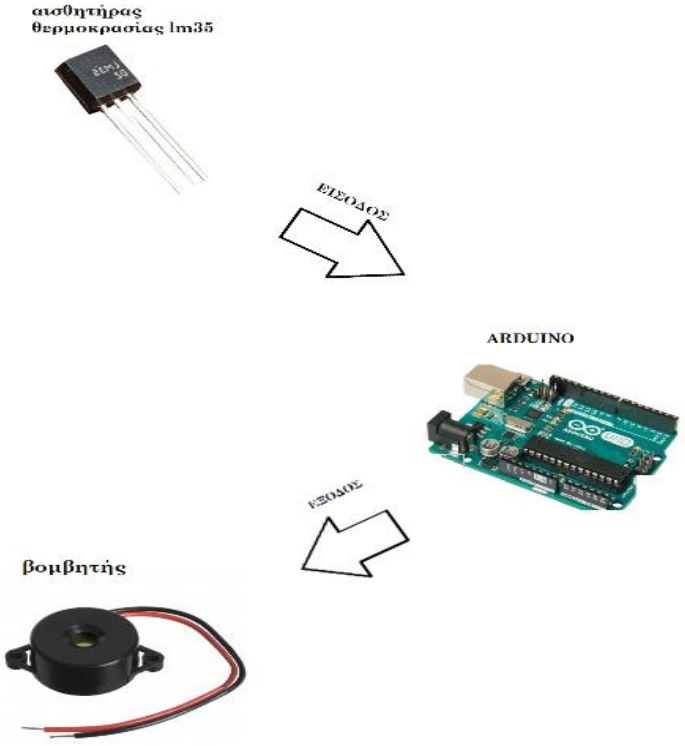
Προφανώς υπάρχουν άλλες λεπτομέρειες στο πιο εξελιγμένο **Arduino** όπως(το μεγαλύτερο **chip** στο **Rev2**, Περισσότερα **Pins** στο **Rev2** και ένα **crypto chip** που δεν θα υπήρχε στο **Rev3**.)

Τώρα που καταλάβαμε διάφορα χαρακτηριστικά από διάφορες πλακέτες και τις δυνατότητες τους,πρέπει να δούμε τα πιο σημαντικά χαρακτηριστικά,στην περίπτωση μας στην πλακέτα **UNO R3**, ώστε να καταλάβουμε πώς να κάνουμε ένα βασικό στήσιμο συστήματος χρησιμοποιώντας της **open source** πλακέτες .

Αρχικά, πρέπει να σκεφτόμαστε μια πλακέτα **Arduino** σαν το σύστημα το οποίο θα επεξεργαστεί πληροφορίες και ενεργεία με ότι εξάρτημα και πηγή θέλουμε(συστήματα εισόδου) και μετά αυτό θα πραγματοποιήσει τις διεργασίες που θα του δώσουμε και ανάλογα, τι ενέργεια και πρόγραμμα έχουμε δώσει στο συστήματα μας να δείξει με ένα εξαρτήματα(συστήματα εξόδου),με λίγα λόγια δίνουμε ενέργεια και μπορούμε να την μετατρέψουμε σε οτιδήποτε θέλουμε,ανάλογα τα εξαρτήματα και τις κινήσεις που θέλουμε να γίνουν.

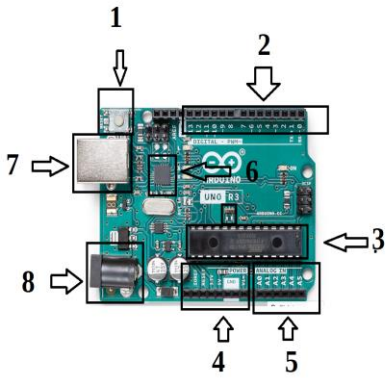
Στο παράδειγμα από κάτω είναι μία ιδέα το πως μπορούμε να σχεδιάσουμε ένα **Arduino** με κατάλληλους εισόδους και εξόδους. Εδώ βλέπουμε για ένα αισθητήρα θερμοκρασίας,όπου μπορεί να ρυθμιστεί,να λαμβάνει την θερμοκρασία θέλουμε μετά να πάνε οι πληροφορίες στο **Arduino** και εμείς θέλουμε να κάνουμε αυτές τι πληροφορίες ως όρια για κάτι ή ουσιαστικά προδιαγραφές,μετα το **Arduino** θα δώσει ενεργεία στον βομβητή,φτιάχνοντας ένα εύκολο και έξυπνο συστήματα ασφάλειας με ήχο για όποια θερμοκρασία υπάρχει .

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156



Εικονα 2.1(Κύκλωμα με Είσοδο και Έξοδο)

Τώρα που καταλάβαμε το πως ένα σύστημα **Arduino** φτιάχνεται και πως είναι ένα ολοκληρωμένο **project**, θα δείξουμε το τι υπάρχει πάνω σε μια πλακέτα ώστε να μας δώσει την ευκαιρία να καταλάβουμε την βασική γνώση και θα μας δώσει να καταλάβουμε πως μπορούμε να κάνουμε ένα δικό μας **project** με ποιο πολύ ευκολία.



Εικονα 2.2(Arduino Uno R3)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Θα αρχίσουμε με κάθε βασικό κομμάτι από την πλακέτα μας που έχουμε αριθμήσει:

1:**Reset**, ένα από τα πιο βασικά κουμπιά της πλακέτας μας,όπου αυτό υπάρχει για το αν θέλουμε να κάνουμε μια επανεκκίνηση το σύστημα.

2:Εδώ βλέπουμε διάφορες θύρες,πρώτα από δεξιά προς αριστερά βλεπουμε το **TX→1** και **RX←0** αυτές οι θύρες υπάρχουν για να στέλνουμε και να λαμβάνουμε δεδομένα,επειδή έχει να κάνει με την σειριακή σύνδεση ,Το ένα έχει να κάνει με την λήψη(**RX**) και το αλλο με την αποστολή δεδομένων(**TX**).Σε συνέχεια, οι υπόλοιπες θύρες έχουν να κάνουν με ψηφιακούς εισόδους και εξόδους τα οποία μπορούμε να τα ρυθμίσουμε εμείς και επίσης θύρες που δίπλα από τον αριθμό τους έχουν το σύμβολο **(~)** ,στην δική μας περίπτωση (~3 ~5 ~6 ~9 ~10 ~11) σχετίζονται με το **PWM** όπου αυτές οι θύρες υποστηρίζουν και αναλογική έξοδο,για παράδειγμα να δούμε την ταχύτητα ενός μοτέρ.

3:Ο βασικός μικροελεγκτής, δηλαδή ο εγκέφαλος της πλακέτας μας οπου από εκεί περνάνε όλα και κάνει τις διεργασίες μας και δίπλα του είναι ο ρυθμιστής που ελέγχει με τι ταχυτητα τρέχει

4:Θύρες ενεργείας,οποί κάθε ένα έχει διαφορετική σημασία, **3,3V** και **5V** είναι και οι δυο θύρες για εξόδου ενέργειας,με τα **Volts** με την δύναμη τους αντίστοιχα, η θύρα **GND** βοηθάει στο να είναι ολοκληρωμένο κύκλωμα(γείωση),όπου όλα τα κυκλώματα χρειάζονται αρνητική και θετική τάση για να λειτουργήσουν αυτό είναι το αρνητικό, **VIN** θύρα για εξωτερικές πηγές ενέργειας που θέλουμε να συνδέσουμε με το **Arduino** αν δεν θέλουμε να χρησιμοποιήσουμε το **USB** ή την υποδοχή ενεργειας.

Τελος, η **IOREF** θύρα σου δίνει την ίδια τάση ενέργειας που έχει το **Arduino**, στην περίπτωση μας **5V** βοηθάει πολύ στο στο να βάζουμε **shields** επειδή την σύνδεση μετάδοσης είναι καλύτερη,

5:**ANALOG IN**, αυτές οι θύρες σχετίζονται για να διαβάζουν αναλογικά σήματα και ανάλογα την τάση του ρεύματος να το μετατρέψουν σε αριθμό από το 0 έως το 1023(πχ **5V**=1023).

6:Για να ανεβάσουμε τον κώδικα μας και για να έχουμε επικοινωνία με τον βασικό μικροελεγκτή, υπάρχει ένας δεύτερος μικροελεγκτής που η χρήση του είναι η επικοινωνία ανάμεσα στον υπολογιστή και το **Arduino[1]**. Όταν για παράδειγμα θα τρέχει ο κώδικας μας, αυτό είναι που στέλνει τα δεδομένα και παίρνει τα δεδομένα μέσα από την **USB** υποδοχή.

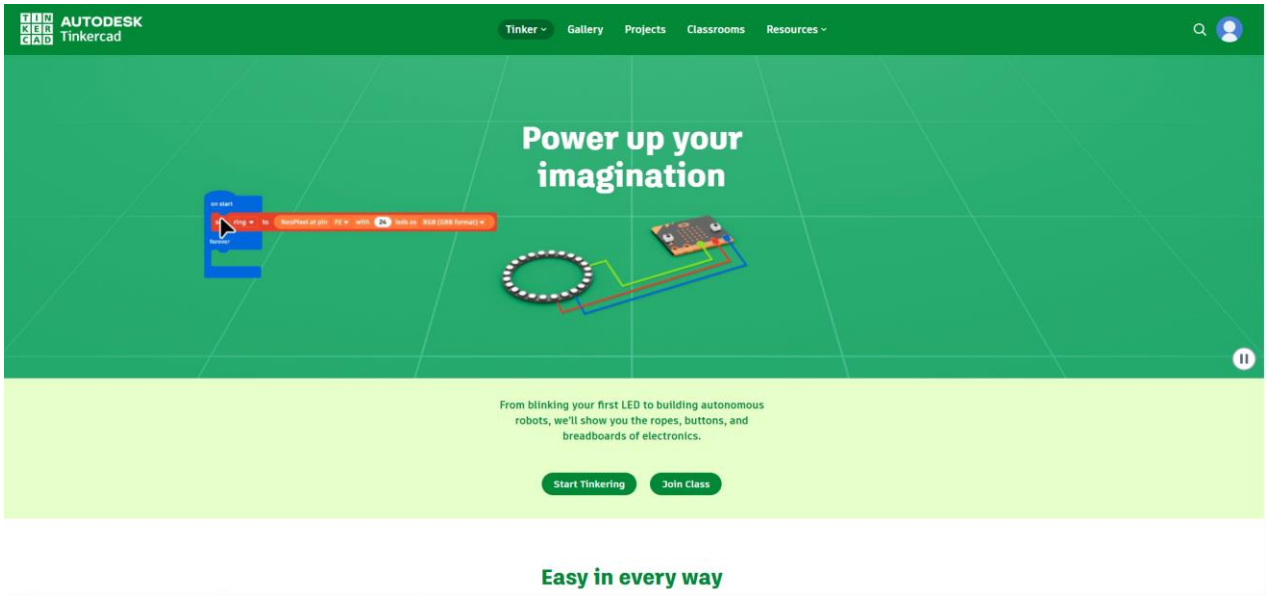
7:**USB** θύρα, έτσι ώστε να δίνουμε τροφοδοσία και να στέλνουμε δεδομένα στο **Arduino**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

8:Υποδοχή τροφοδοσίας, ενώ έχουμε και άλλες πηγές ενέργειας μπορούμε να έχουμε επίσης ενέργεια από ένα εξωτερικό τροφοδοτικό περίπου **7-12V**

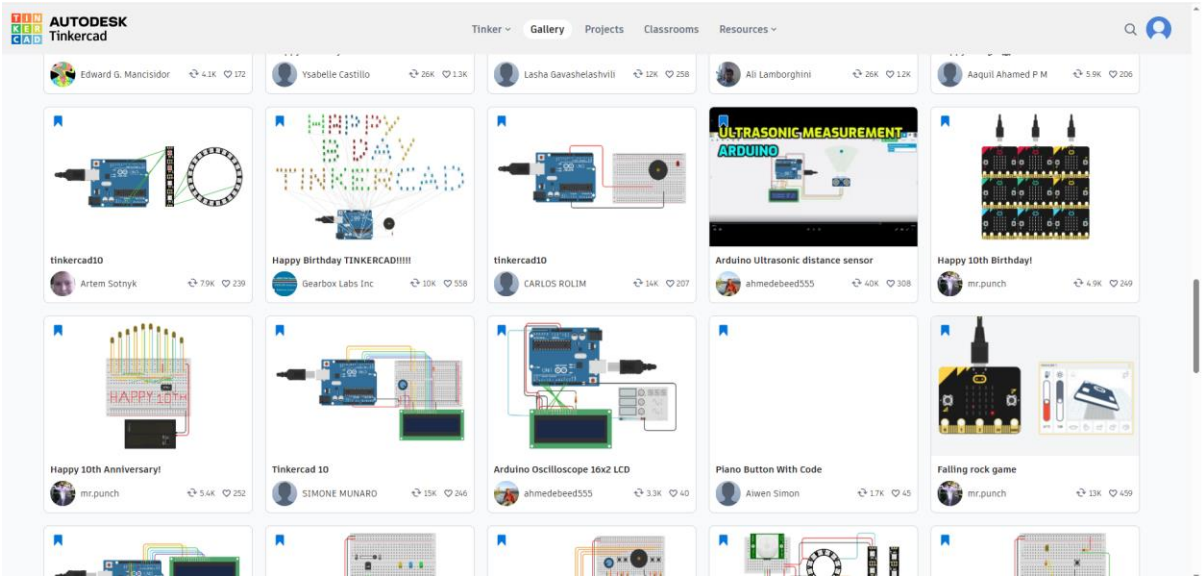
Τώρα που μάθαμε την λογική της πλακέτας,διάφορων εξαρτημάτων και το τι είναι ένα σύστημα **Arduino**, θα δούμε το πως μπορούμε να κάνουμε μια προσομοίωση και σχεδίαση ενός κυκλώματος, μέσα από ψηφιακά μέσα. Επειδή προφανώς δεν έχει κάθε άτομο την δυνατότητα και την πρόσβαση σε πλακέτες και εξαρτήματα, μπορούμε πάρα πολύ εύκολα και πάλι να προγραμματίσουμε και να μάθουμε παραπάνω **Arduino** μέσα από μερικές προσβάσιμες και εύκολες πλατφόρμες. Αυτές που θα δούμε μαζί είναι το: **Tinkercad Circuits,Wokwi, SimulIDE**

Tinkercad Circuits:Είναι μια από τις πλατφόρμες που θα χρησιμοποιούμε,όπου σε αυτή την πλατφόρμα την σχεδιάζουμε και κάνουμε προσομοίωση κυκλωμάτων **Arduino**, με τα διάφορα εξαρτήματα που υπάρχουν στην διαθεσιμότητα μας, επίσης μέσα από την πλατφόρμα μπορούμε να μάθουμε τα κυκλώματα και τα εξαρτήματα πιο εύκολα και να τα ρυθμίσουμε με τον τρόπο που θέλουμε και τις δυνάμεις που χρειαζομαστε. Είναι γενικά ένας εύκολος τρόπος να μάθουμε να χρησιμοποιούμε ένα **Arduino** και να το προγραμματίσουμε με τον τρόπο που θέλουμε,επίσης έχει πολλά έτοιμα projects όπου μπορούμε να μάθουμε και ουσιαστικά να ξεκινήσουμε με τα βασικά για να καταλάβουμε τα αρχικά στάδια ενός κυκλώματος. Γενικά, είναι ένα πολύ ασφαλές περιβάλλον το οποίο θα χρησιμοποιήσουμε να μάθουμε **Arduino**, γιατί και μέσα από τα κυκλώματα τα έτοιμα η όχι,έχει την επιλογή να ρωτήσουμε για κάθε εξάρτημα που υπάρχει και να δούμε τις δυνατότητες του. Επίσης, είναι προσβάσιμο,επειδή με κάθε προφίλ που φτιαχτούμε παραμένουν στο cloud και μπορούμε να ξαναδούμε και να ξαναφτιάξουμε οτι και όποτε χρειάζεται



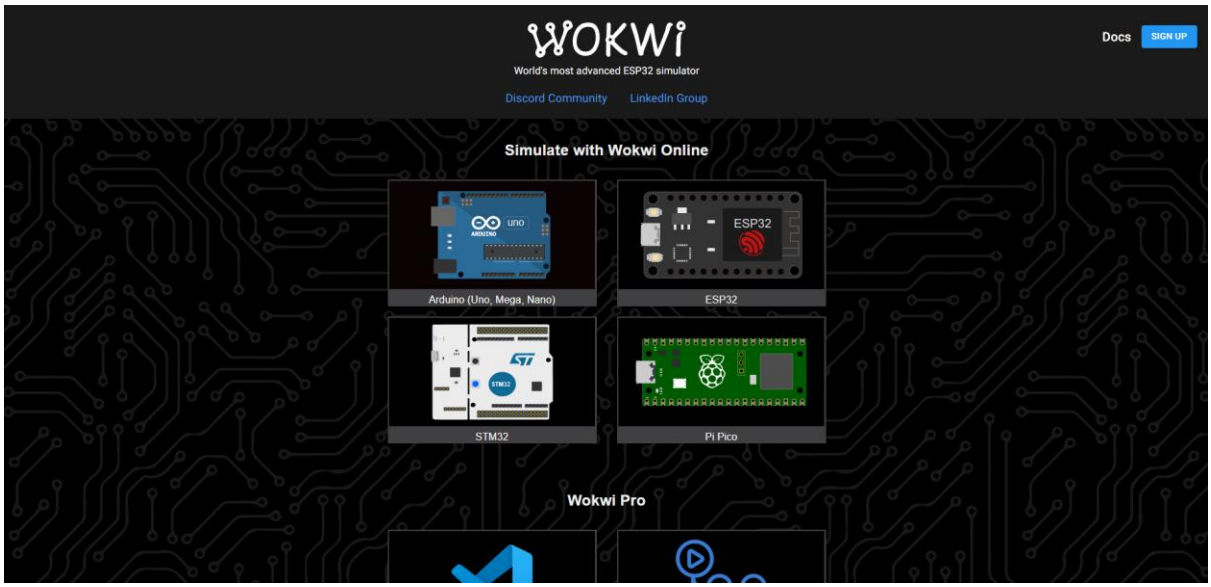
Εικόνα 2.3(https://www.tinkercad.com/)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156



Εικόνα 2.4(https://www.tinkercad.com/)

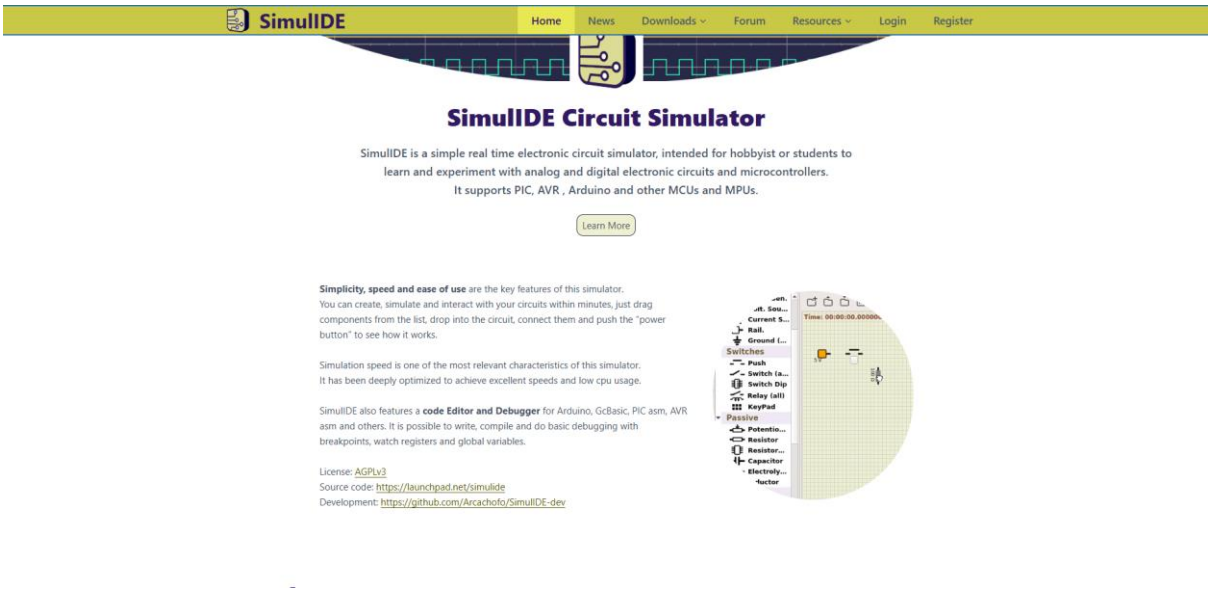
Wokwi:Είναι κάτι παρόμοιο με το **Tinkercad** αλλά ακόμα πιο εξελιγμένο, σε αυτό μπορούμε να το τροποποιήσουμε να γράφεις τον κώδικα μέσα από **Visual Studio**, μπορούμε να γράψουμε και με **python** τα projects μας και πιο γρήγορα, γιατί ουσιαστικά δεν έχει πολύ διαδικασία για να ξεκινήσουμε να γράψουμε και να σχεδιάσουμε ένα κύκλωμα σαν στο **Tinkercad**.Είναι, καλύτερη αυτή η σχεδίαση κυκλωμάτων για άτομα που έχουν παραπάνω εμπειρία,δίνει παραπάνω ευκαιρίες γιατί έχει πιο πολλά εξαρτήματα και πιο πολλές πλακέτες και επιλογές προσομοίωσης δικτύου



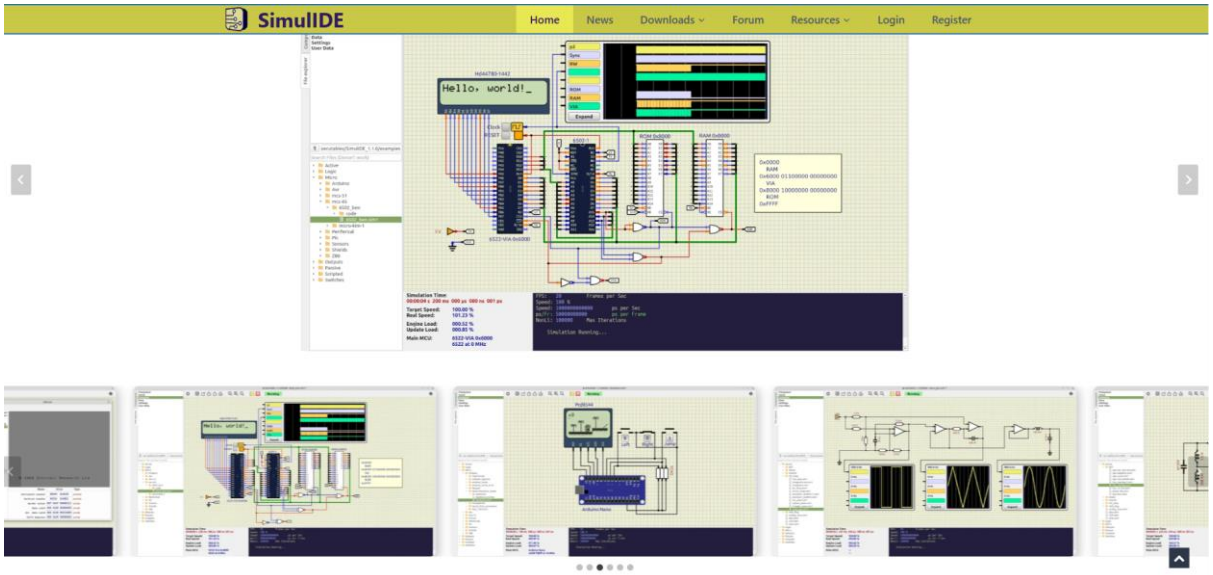
Εικόνα 2.5(https://wokwi.com/)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

SimulIDE:Είναι μια πλατφόρμα offline κυκλωμάτων **Arduino**,άρα πρέπει να το κατεβάσουμε. Προτείνεται σε άτομα που θέλουν να δουν το φυσικό επίπεδο των κυκλωμάτων και εξαρτημάτων,μπορεί επίσης να πάρει δεδομένα και κώδικα κατευθείαν από την **Arduino IDE**.



ΕΙΚΟΝΑ 2.6 (https://simulide.com)

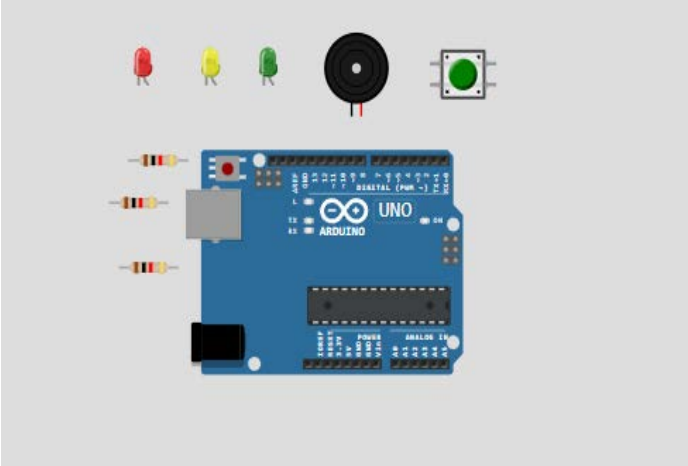


Εικονα 2.7(https://simulide.com)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

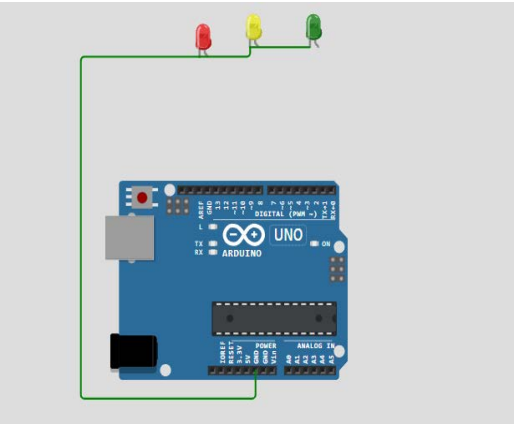
ΚΕΦΆΛΑΙΟ 3 Πρώτη σχεδίαση

Λοιπόν, αφού δείξαμε τις πλατφόρμες και τα στοιχεία τους θα αρχίσουμε με την πρώτη σχεδίαση στο **Wokwi**.Επιλέξαμε αυτή την πλατφόρμα για την μεγαλύτερη προσβασιμότητα και γενικά επειδή θέλουμε να γράψουμε τον κώδικα μας ,αντί στο **Tinkercad** όπου έχει μόνο προγραμματισμό με **Blocks**. Αρχικά, πρέπει να πάρουμε μια ιδέα και να το σχεδιάσουμε και σε κώδικα άλλα και τα εξαρτήματα του,στην περίπτωση μας,θα κάνουμε ένα **Arduino** το οποίο λειτουργεί ως αυτόματο φανάρι δρόμου και θα έχει επίσης σύστημα με κουμπί, όπου όταν το πατάμε θα γίνεται το φανάρι κόκκινο και θα βγάζει ένα ήχο ώστε να υπάρχει η ένδειξη για αυτό,ουσιαστικά ένα σύστημα κυκλοφορίας για πεζούς και οδηγούς. Πρώτα, θα δείξουμε τα εξαρτήματα που θα χρησιμοποιήσουμε τα οποία θα είναι στην εικόνα μας



Εικόνα 3.0(Εξαρτήματα του project 1)

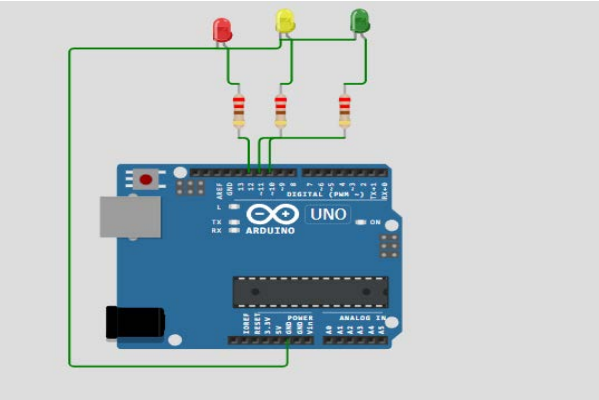
Στο κύκλωμα μας θα χρησιμοποιήσουμε 3 **LED** και 3 **resistors**,1 **Buzzer** και 1 **Pushbutton**.Πρώτα θα γίνουν οι συνδέσεις των **LED**, ανοιχτού κυκλώματος, άρα μόνο **GND**.χωρις **PIN**



Εικόνα 3.1(Σύνδεση **LED** στην υποδοχή **GND**)

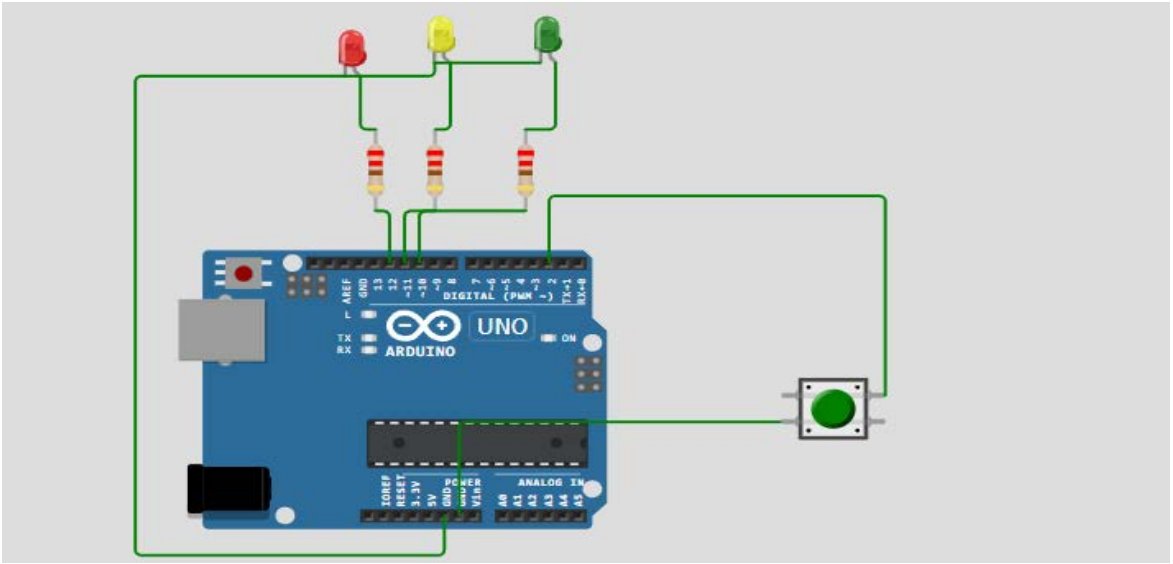
Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Αυτό που παρατηρούμε στο ανοιχτό κύκλωμα μας,συνδέσαμε όλα τα **LED** με το ίδιο καλώδιο έτσι ώστε να μην χρησιμοποιήσουμε **Breadboard** όπου θα έκανε λιγότερες τις συνδέσεις μας,αλλά αυτό είναι ένα πλεονέκτημα του **Wokwi**. Μετά θα βάλουμε 3 **resistors** έτσι ώστε να γίνει το κύκλωμα μας κλειστό και να υπάρχει τάση ρεύματος,χωρίς τα **LED** να υπερθερμανθούν,με **220Ω** Units, σε οποία **PIN** θέλουμε



Εικόνα 3.2(Σύνδεση **LED** σε **PIN** και **Resistors** (**PIN** 12-11-10)

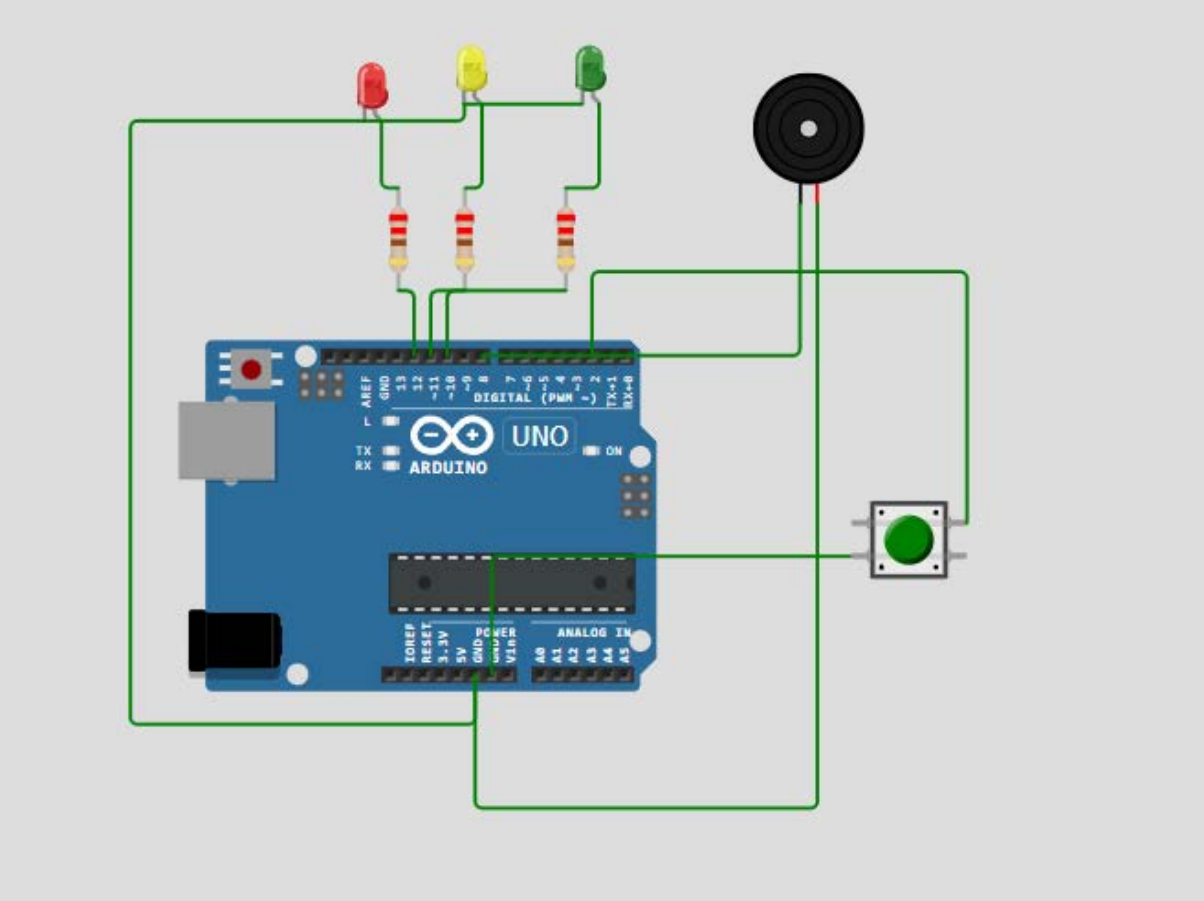
Το επόμενο βήμα μας είναι το κουμπί το οποίο θα χρησιμεύει στην αλλαγή του **LED** και του **Buzzer**,στην περίπτωση μας δεν θα βάλουμε resistor,επειδή θα κάνουμε αλλαγές στον κώδικα μας έτσι ώστε να μην χρειάζεται και επίσης πρέπει να καταλάβουμε λίγο την μεθοδολογία ενός **Pushbutton** πριν το εντάξουμε στο κύκλωμα μας.Για παράδειγμα το **Pushbutton** έχει 4 ποδαράκια (1-2-3-4),τα οποία ενώνονται μεταξύ τους σε ομάδες(1-2 και 3-4)αντίστοιχα όταν δεν πατιέται το κουμπί,και όταν ενεργοποιείτε οι ομάδες ενώνονται δημιουργώντας ένα κλειστό κύκλωμα,οπότε στο κύκλωμα μας η μια ομάδα θα είναι προς το **PIN** και η άλλη θα είναι στο **GND** έτσι ώστε για να υπάρχει σωστή τροφοδοσία.



Εικόνα 3.3(Σύνδεση **Pushbutton** με το **Arduino**(**PIN** 2 **AND GND**)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Και τέλος θα ενώσουμε το **buzzer** μας όπου θα βγάζει ήχο μόνο όταν πατάμε το **Pushbutton**



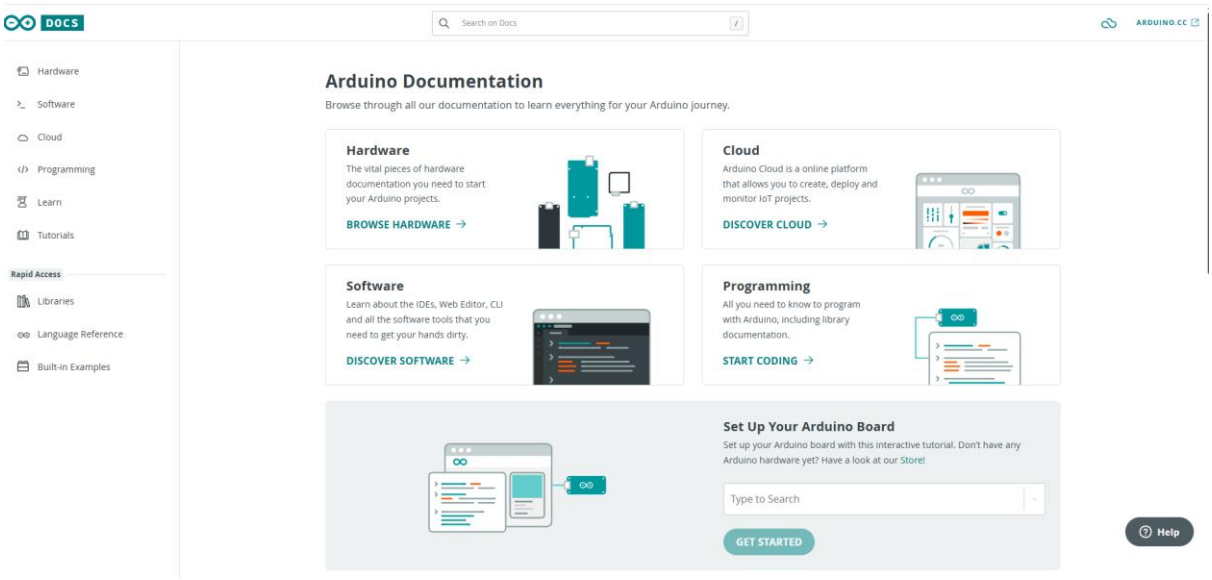
Εικόνα 3.4(Σύνδεση **Buzzer** με το **Arduino**(**PIN** 8 **AND GND**)

Και τώρα που ολοκληρώσαμε το κύκλωμα μας,πρέπει να το προγραμματίσουμε έτσι ώστε να κάνουμε τις διαδικασίες που θέλουμε.Πρώτα πρέπει να ξέρουμε ότι γράφουμε σε κώδικα **Arduino Sketch** ουσιαστικά μια μορφη της **C++** με μοναδικές του συναρτήσεις και αλλους κωδικες απο **Arduino IDE**.Στην περίπτωση μας θα χρησιμοποιήσουμε**(milis(), pinMode(), digitalRead(),digitalWrite, tone(), noTone(), Serial.begin(), Serial.println())**.Όλα αυτά τα παραπάνω θα δούμε μέσα στον κώδικα μας το τι κάνουν και επίσης που τα χρησιμοιπιούμε.

Επίσης θα τα τρέξουμε μέσα από το **Wokwi** και δεν χρειάζεται να έχουμε κάποιο άλλο **tool**

Όλες οι συναρτήσεις του **Arduino** μπορούμε να τα βρούμε πολύ εύκολα σε όλο το δίκτυο,επειδή είναι **open source** όπως αναφέραμε αλλά κυρίως σε αυτό τον ιστότοπο πιο συγκεκριμένα στην ιστοσελίδα του **Arduino**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156



Εικόνα 3.5 με συναρτήσεις από το official site του **Arduino**)
https://docs.arduino.cc/language-reference/#functions

Επίσης στην ίδια ιστοσελίδα μπορούμε να δούμε κάθε πράγμα που σχετίζεται με το **Arduino** από πλακέτες μέχρι το λογισμικό που μπορούμε να χρησιμοποιήσουμε για να κάνουμε τον προγραμματισμό μας, μαθήματα και παραδείγματα για να εξασκηθούμε, το **cloud** το οποίο μπορούμε να ανεβάσουμε το κώδικα μας έτσι ώστε να το αποθηκεύσουμε για το μέλλον ή για τρόπο μεταφοράς κώδικα σε άλλα άτομα.Γενικά,υπάρχουν πολλές ευκαιρίες και μέσα από τα social media και την ομάδα τους και από άλλα άτομα που χρησιμοποιούν **Arduino** να μάθει κάποιος και να προοδεύσει το **project** η να πάρει βοήθεια σε κάτι μέσα από την ιστοσελίδα.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Τώρα θα πάρουμε βήμα προς βήμα τον κώδικα μας έτσι ώστε να το καταλάβουμε καλύτερα:

```
1
2   const int redLED    = 12;
3   const int yellowLED = 11;
4   const int greenLED  = 10;
5   const int buttonPin = 2;
6   const int buzzerPin = 8;
```

Εικόνα 3.6(Κώδικας από 1-6)

1-6:Δηλώνουμε τα εξαρτήματα που έχουμε στα **Pin** τους,τα **Led(red,yellow,green)**, το **button**, το **buzzer**.

```
8
9   bool buttonPressed     = false;
10  bool buttonHandled     = false;
11  unsigned long buttonPressTime    = 0;
12  unsigned long forcedRedStartTime = 0;
13
```

Εικόνα 3.7(Κώδικας από 9-12)

9-12:μας δείχνει ότι έχει πατηθεί το κουμπί αλλά δεν είναι ακόμα κόκκινο,10:μας δείχνει ότι έχει πατηθεί και είναι κόκκινο το **LED+Buzzer**,έτσι ώστε να μην ξαναγίνει και να αρχίσει η επανάληψη, 11:στιγμή καταγραφής για το πότε πατήσαμε το κουμπί, 12:στιγμή που άναψε το κόκκινο **Led** και το **Buzzer**.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

```
15   const unsigned long waitTime          = 3000;
16   const unsigned long forcedRedDuration = 4000;
17   unsigned long lastChangeTime          = 0;
18   const unsigned long trafficInterval   = 4000;
19
20   int trafficState = 0;
```

Εικόνα 3.8(Κώδικας 15-20)

15-20: 15: το πόσο θα περιμένουμε μέχρι να ανάψει το κόκκινο μετά το πάτημα του button, 16:το πόσο κρατάει το **Led** να είναι κόκκινο και το **Buzzer** να κάνει ήχο, 17:βλέπει το πότε έγινε αλλαγή στην κανονική ροή των **Led**, 18:Μετράει την διάρκεια ενός κύκλου των **Led**, 20:κρατάει σε ποια φάση χρώματος είναι.

```
22   void setup() {
23     pinMode(redLED,    OUTPUT);
24     pinMode(yellowLED, OUTPUT);
25     pinMode(greenLED,  OUTPUT);
26     pinMode(buttonPin, INPUT_PULLUP);
27     pinMode(buzzerPin, OUTPUT);
28     Serial.begin(9600);
29
30     trafficGreen();
31     lastChangeTime = millis();
32   }
```

Εικόνα 3.9(Κώδικας από 22-32)

22-32: 23-24-25-27 Ρυθμίζουμε τα **Led** και το **Buzzer** για την έξοδο,δηλαδή την λειτουργία τους, 26:φτιάξαμε μια εσωτερική αντίσταση **pull up-** έτσι ώστε να μην βάλουν αντίσταση στο κύκλωμα μας και το **Button** να είναι **HIGH** όταν δεν το πατάμε , 28:έναρξη σειριακής επικοινωνίας(πιο πολύ για **debugging**), 30:κάνουμε το **Led** πράσινο κατευθείαν, 31:καταγράφουμε την εκκίνηση του κανονικού χρόνου.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

```
34  void loop() {
35    unsigned long now = millis();
36
37    // 1) Διαβάζουμε το κουμπί
38    if (digitalRead(buttonPin) == LOW && !buttonPressed) {
39      buttonPressed    = true;
40      buttonPressTime  = now;
41      Serial.println("Button Pressed!");
42    }
```

Εικόνα 3.10(Κώδικας από 34-42)

34-42:35:αποθηκεύουμε την επανάληψη,38:αν το κουμπί πατήθηκε τότε και για την μια φορά που το πατήσαμε, 39-40:θέτουμε το πάτημα ότι έγινε,αποθηκεύουμε τον χρόνο και εκτυπώνουμε το μήνυμα ότι πατήθηκε.

```
45  if (buttonPressed && !buttonHandled && now - buttonPressTime >= waitTime) {
46    trafficRed();
47    tone(buzzerPin, 200);
48    buttonHandled     = true;
49    forcedRedStartTime = now;
50    Serial.println("Forced Red + Buzzer ON");
51  }
```

Εικόνα 3.11(Κώδικας από 45-51)

45-51:Μόλις πατήσουμε το κουμπί θα περιμένει **3s** από το πάτημα,μετά θα ανάψει το κόκκινο **Led** και θα βγάλει ήχο το **Buzzer** στα **200 Hz**,μετά αρχίζει να μετράει τον χρόνο του κόκκινου **Led.**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

```
54  if (buttonPressed && buttonHandled && now - forcedRedStartTime >= forcedRedDuration) {
55    noTone(buzzerPin);
56    buttonPressed = false;
57    buttonHandled = false;
58    trafficState  = 0;
59    trafficGreen();
60    lastChangeTime = now;
61    Serial.println("Forced Red OFF — Back to Normal");
62  }
```

Εικόνα 3.12(Κώδικας από 54-62)

54-62:Μόλις τελειώσουν τα **4s** του αναμένουν κόκκινου **Led**, σταματάει τον ήχο ,επαναφέρει τα πάντα στο αρχικό στάδιο και ανάβει το πράσινο **Led.**

```
  if (!buttonPressed) {
    if (now - lastChangeTime >= trafficInterval) {
      lastChangeTime = now;
      nextTrafficState();
    }
  }
}
```

Εικόνα 3.13(Κώδικας 64-71)

64-71:Γίνεται η εκτέλεση του μόνο όταν δεν έχουμε κόκκινο **Led**, και μετά από **4s** καλεί το επόμενο βήμα

```
74  void nextTrafficState() {
75    trafficState++;
76    if (trafficState > 2) trafficState = 0;
77    if (trafficState == 0)      trafficGreen();
78    else if (trafficState == 1) trafficYellow();
79    else                        trafficRed();
80  }
```

Εικόνα 3.14(Κώδικας 74-80)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

74-80:Κάνει την κίνηση των **Led** ανάλογα την τιμή τους και τα καλεί ανάλογα την συνάρτηση που πέφτει ,0→1→2→0→1→2.

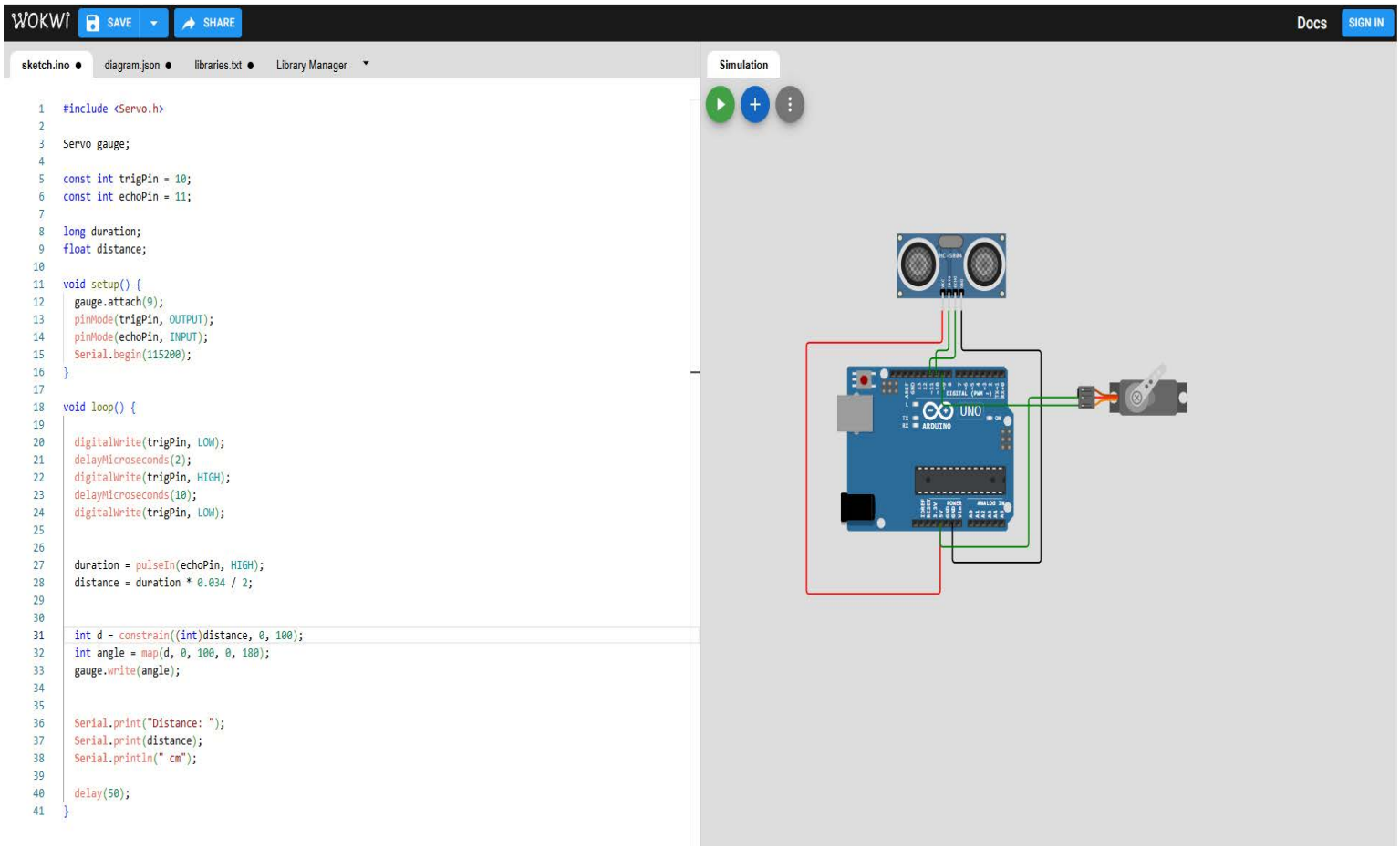
```
82  void trafficGreen() {
83    digitalWrite(greenLED, HIGH);
84    digitalWrite(yellowLED, LOW);
85    digitalWrite(redLED, LOW);
86  }
87
88  void trafficYellow() {
89    digitalWrite(greenLED, LOW);
90    digitalWrite(yellowLED, HIGH);
91    digitalWrite(redLED, LOW);
92  }
93
94  void trafficRed() {
95    digitalWrite(greenLED, LOW);
96    digitalWrite(yellowLED, LOW);
97    digitalWrite(redLED, HIGH);
98  }
```

Εικόνα 3.15(Κώδικας 82-98)

82-98:Ορίζουμε τις εξόδους για τα **Led**,και κάνουν κανονικό κύκλο αν δεν πατήσουμε το **Button**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΚΕΦΆΛΑΙΟ 4 Δεύτερη σχεδίαση



Εικόνα 4.0(Σχεδίαση 2 και κώδικας)

Πρώτα θα αρχίσουμε με το κύκλωμα μας και το πως το συνδέσαμε, όπως καταλάβαμε και από το προηγούμενο διάγραμμα, τα εξαρτήματα μας χρειάζονται πολλές συνδέσεις,και στην περίπτωση μας οπου εδώ έχουμε ένα **Servo** και **Distance Sensor** θα χρησιμοποιηθούν αρκετά καλώδια,άλλα το καλό με την ψηφιακή πλατφόρμα είναι ότι μπορούμε να τα κάνουμε όλα πολύ γρήγορα και εύκολα.Το **Distance Sensor** εχει 4 **Pin** τα δυο για τάση (**GND-VCC**) και τα άλλα δυο είναι ψηφιακά (**TRIG -ECHO**),Το **Servo** χρειάζεται απλά ένωση το **GND** με το **V+** και το **Pwn** να μπει σε κάποια ψηφιακή θήρα

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

```
1   #include <Servo.h>
2
3   Servo gauge;
4
5   const int trigPin = 10;
6   const int echoPin = 11;
7
8   long duration;
9   float distance;
10
11  void setup() {
12    gauge.attach(9);
13    pinMode(trigPin, OUTPUT);
14    pinMode(echoPin, INPUT);
15    Serial.begin(115200);
16  }
```

Εικόνα 4.1(Κώδικας 1-16)

Φορτώνουμε την βιβλιοθήκη **Servo** για να το χειριστούμε, μετά δημιουργούμε το αντικείμενο έτσι ώστε για να το ελέγχουμε. Μετα,καθορίζουμε τις ψηφιακές υποδοχές μας και φτιάχνουμε μεταβλητή για να μετράμε την διάρκεια του **echo** σε **ms**, και μια άλλη μεταβλητή για να καθορίζουμε την απόσταση σε εκατοστα.Μετά συνδέουμε το **Servo**, το **Trig** θα είναι έξοδος και το **Echo** είσοδος και και μετά ξεκινάμε μια σειριακή επικοινωνία κυρίως για **debugging**

```
17
18  void loop() {
19
20    digitalWrite(trigPin, LOW);
21    delayMicroseconds(2);
22    digitalWrite(trigPin, HIGH);
23    delayMicroseconds(10);
24    digitalWrite(trigPin, LOW);
25
26
```

Εικόνα 4.2(Κώδικας 18-24)

Εδώ κάνουμε το **Trig Low** για **2 ms** έτσι ώστε να είναι εύκολο το ξεκίνημα του, μετά το κάνουμε **High** για **10 ms** για να αποστείλει τα σήματα ήχου και μετά το κάνουμε **Low** για να περιμένουμε το **Echo**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

```
25
26
27      duration = pulseIn(echoPin, HIGH);
28      distance = duration * 0.034 / 2;
29
```

Εικόνα 4.3(Κώδικας 27-28)

Αφού στείλαμε τον ήχο μας,το **Echo** μετράει σε **ms** πόσο διαρκεί ο παλμός και όταν γυρίσουν με την ταχύτητα του ήχου και πραγματοποιείται διαίρεση με το δυο θα δούμε την απόσταση του ήχου σε εκατοστά

```
31      int d = constrain((int)distance, 0, 100);
32      int angle = map(d, 0, 100, 0, 180);
33      gauge.write(angle);
34
35
36      Serial.print("Distance: ");
37      Serial.print(distance);
38      Serial.println(" cm");
39
40      delay(50);
41  }
```

Εικόνα 4.4(Κώδικας 31-41)

Μέτα μετατρέπουμε την απόσταση σε ακέραιο,κάνουμε την τιμή να μην περάσει τα όρια που θέσαμε,αυτά τα όρια που θέσαμε αντιστοιχούν γραμμικά στης γωνιες που θα κάνει το **Servo** και μετά εκτελούμαι την εντολή για να κάνει την κίνηση που θέλουμε

Στέλνουμε την τιμή της απόστασης πίσω σε εμάς για έλεγχο και μετά κάνουμε μια παύση στο **Servo** για **50ms** για να είναι πιο ομαλό

Γενικά η πλατφόρμα του **Wokwi[3]** είναι πάρα πολύ καλή για να εμπεδώσεις τις γνώσεις σου καλύτερα και να αρχίσεις να κάνεις κυκλώματα στα γρήγορα χωρίς να υπάρχει κάποιος κίνδυνος η να μην υπάρχει η δυνατότητα στην δυσκολία των φυσικών κυκλωμάτων.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΚΕΦΆΛΑΙΟ 5 Συμπεράσματα και Γλώσσα Arduino

Από ότι είδαμε η μορφή της ηλεκτρονικής είναι ένας ωραίος τρόπος να γίνεις πιο δημιουργικός με τον τρόπο σκέψης και δεν χρειάζεται τα φυσικά μέσα,είδαμε μαζί μερικές πλατφόρμες αλλά όχι όλες,υπάρχει και άλλη μια που είναι κάτι ενδιάμεσο στο **Wokwi** και το **Tinkercard[4]**, είναι το **Fritzing**. Η ηλεκτρονική είναι κάτι που όλοι θα έπρεπε να ξέρουμε έστω και λίγο και υπάρχει η δυνατότητα της εξέλιξης και σε άλλες πλακέτες με ακόμα πιο καινούργια εξαρτήματα για να καταφέρουμε μεγαλύτερα πράγματα,για παράδειγμα τα **Raspberry** που αναφέραμε στην αρχή,ένα κάτι ενδιάμεσο πιστεύω στην σκέτη ηλεκτρονική και πληροφορική, έχει πιο πολλά tools και δύναμη στο να αντέχει αυτά που ζητάμε. Υπάρχουν άφθονα παραδείγματα,αλλά για το μέλλον πιστεύω ότι ο κόσμος έχει ξεκινήσει να καταλαβαίνει την δύναμη ενός μικροϋπολογιστή τι μπορεί να καταφέρει και τι να δείξει,είτε αυτό είναι για την καθημερινότητα η για εκπαίδευση,πάντα υπάρχει κάτι να δείξουμε από τους μικρουπολογιστές. Η ηλεκτρονική έχει τεράστιο επίπεδο εξελίξεις και ότι ακουμπήσει στο έργο μας δεν είναι ούτε το δέκα της εκατό της ύλης,θα μπορούσαμε να γράφαμε μέρες για κάθε εξάρτημα και κάθε **Pin**

Θετικά:Μερικά από τα θετικά είναι ότι όλες οι πλατφόρμες μας είναι πάρα πολύ προσβάσιμες, και μπορούμε όταν έχουμε κάποιο σχέδιο στην διαθεσιμότητα μας να το προγραμματίσουμε και έχουμε εύκολη πρόσβαση στην υλοποίηση του. Μερικές απο αυτές είναι το **SimulIDE[5]** όπου είναι από της λίγες πλατφόρμες που είναι **offline** και δεν χρειάζεται σύνδεση στο διαδίκτυο για να κάνουμε τα σχέδια ή να δούμε διάφορα εξαρτήματα και πλακέτες. Δευτερον,όλα τα σχέδια μας δεν έχουν τρόπο να πάθουν βραχυκύκλωσης με κάποια τάση ρεύματος και υπάρχουν τρόποι όπου οι πλατφόρμες μας δείχνουν το πρόβλημα του σχεδίου μας ή το πως να το συνδέσουμε σωστά με την πλακέτα και τι εξαρτήματα μπορούμε να χρησιμοποιήσουμε μέσα από την συλλογή του κάθε εργαλείου όπου θέλουμε να κάνουμε το σχέδιο μας. Επίσης,μέσα από τις πλατφόρμες μπορούμε να μάθουμε τα βασικά στοιχειά σχεδιασμού **Arduino** για παράδειγμα το **Tinkercad** έχει διάφορα αρχάρια παραδείγματα για να μπούμε στην ιδεολογία του σχεδιασμού και βήμα βήμα να μας καθοδηγούν στο αποτελέσματα του **project** που μας δίνει και στην τελική να προσθέσουμε ή να αφαιρέσουμε πράγματα για να καταλάβουμε την λογική τους.

Αρνητικά:Ενώ οι πλατφόρμες που έχουμε,μπορούν να μας δείξουν και να μας διευκολύνουν πολύ με την χρήση τους,εμείς αν θέλουμε να κάνουμε κάποιο **Arduino Project** σε φυσική μορφή πρέπει να καταλάβουμε ότι το κύκλωμα μας είναι εντελώς πιο ευάλωτο και ότι δεν έχουμε τόσες επιλογές και περιπτώσεις για λάθοι,πολλά πράγματα που δεν είχαμε υπόψιν πριν πρέπει τότε να δούμε καλύτερα, για παράδειγμα την σύνδεση καλωδίων και της τάσεις ρεύματος όπου κάποιο λάθος μπορεί να προκαλέσει στην καταστροφή του κυκλώματος μας και μπορεί και των εξαρτημάτων μας .Άλλο ένα αρνητικό είναι ότι οι πλατφόρμες μπορούν να μην έχουν κάθε εξάρτημα που θέλουμε η το συγκεκριμένο που ψάχνουμε ,για παράδειγμα στο πρότζεκτ που κάναμε έκανα διάφορες αλλαγές επειδή δεν έβρισκα κατάλληλα **tools**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

και επεκτάσεις που είχα στο μυαλό ενώ είναι γνωστά στην φυσική μορφή τους, δηλαδή μερικές από τις πλατφόρμες δεν μας δίνουν ολόκληρη την ελευθερία που έχουμε όπως είναι να αγοράσουμε όλα τα εξαρτήματα που υπάρχουν.Επίσης το να μάθεις ουσιαστικά **Arduino** είναι βασισμένο στην θέληση σου στο πόσο θέλεις να μάθεις,οι περισσότερες από τις πλατφόρμες δεν έχουν καθαρή εξήγηση στο τι μπορείς να κάνεις και δεν υπάρχουν καθαρά βήματα στο πως μπορείς να εξελιχθείς και να φτιάξεις κάτι που θέλεις, υπάρχει ένα μεγάλο βήμα από το αρχάριο μέχρι το κανονικό επίπεδο χρήστη.Επίσης η γλώσσα που χρησιμοποιήσουμε για να γράψουμε το σχεδιάγραμμα μας δεν υπάρχει σε κάποια από τις πλατφόρμες για να μάθουμε,είτε θα δούμε από άλλους χρήστες η θα το ανακαλύψουμε μόνοι μας κάτι που μπορεί να απωθήσει κάποιον από το να ξεκινήσει κάτι που θέλει και όταν πρέπει να χρησιμοποιήσουμε κάποιο εξάρτημα πρέπει να δούμε και να μάθουμε με ποια πρόταση κωδικά και ποια συνάρτηση πρέπει να καλέσουμε έτσι ώστε να χρησιμοποιηθεί σωστά.

Γενικά από ότι έχουμε δει υπάρχουν πολλά θετικά και αρνητικά με την χρήση των ψηφιακών μέσων που έχουμε, άλλα προφανώς είναι βασισμένο στο πόσο ο χρήστης θέλει να δώσει ευκαιρία στο να μάθει και να σχεδιάσει το δικό του μοναδικό κύκλωμα αλλά δεν μπορεί να αψηφήσει κανένας την ευκολία και την πρόσβαση σε αυτά που είδαμε. Το πιο δύσκολο από όλα είναι γενικά ο κώδικας και το πως μπορούμε να μάθουμε να γράφουμε το πρότζεκτ που θέλουμε,και για αυτό εμείς στο επόμενο κεφάλαιο θα δείξουμε της βασικές αρχές του **Arduino Sketch** έτσι ώστε να μπορεί ο χρήστης να έχει μια ευκαιρία στο να διευκολύνει το ταξίδι του στον αυτοματισμό,με διάφορα παραδείγματα πάνω στην γλώσσα και μετά θα έχουμε ένα λυσάρι για να μπορούμε να κατανοήσουμε την γλώσσα.

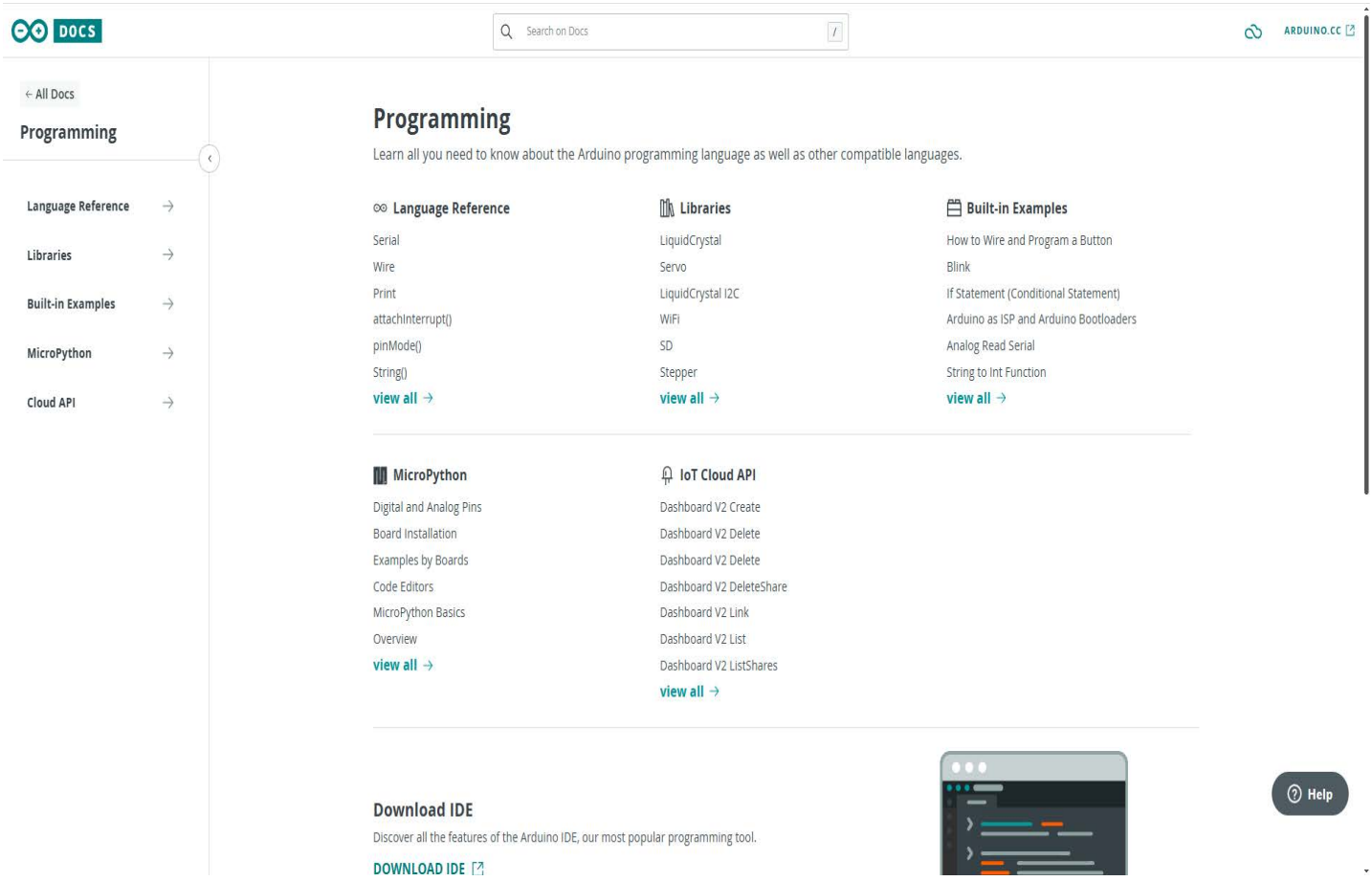
Τώρα που μάθαμε τι είναι το **Arduino** και δείξαμε τα παραδείγματα μας και στο τέλος συγκρίναμε και είδαμε της ψηφιακές πλατφόρμες όπου δείξαμε και τα παραδείγματα μας,θα μπούμε στην λογική της γλώσσας το **Arduino** για να καταλάβουμε μερικές συναρτήσεις και τα βασικά έτσι ώστε να μπορούμε να κατανοήσουμε πως να γράψουμε ένα πρότζεκτ.

Λοιπόν θα ξεκινήσουμε με τα βασικά,η γλώσσα που γράφουμε Arduino βασίζεται στην **C++** και παρόλο που φαίνεται σαν άλλη γλώσσα ότι πρόγραμμα Arduino μεταγλωττίζεται σε **C++** και μετά σε μηχανικό κώδικα για να εκτελεστεί από τον μικροελεγκτή της πλακέτας,οπότε η γλώσσα είναι μια ποιο απλή **C++** και χρησιμοποιεί παραπάνω βιβλιοθήκες και συναρτήσεις φτιαγμένες για ειδικές χρήσεις με φυσικό υλικό όπως δείξαμε και στο παράδειγμα.

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Πρώτα από όλα ότι υπάρχει στην **C++** υπάρχει και στο **Arduino** για παράδειγμα πολλά πράγματα που θα έπρεπε να γράψουμε στην **C++**,στην γλώσσα του **Arduino** υπάρχουν είδη και είναι ποιο εύκολα να τα καλέσουμε. Στην **C++** θα έπρεπε να χρησιμοποιούμε διάφορες βιβλιοθήκες για να κάνουμε παρομοια πράγματα με ότι έχουμε σε ένα **Arduino** όπου αυτές είναι είδη ενσωματωμένες επίσης και άλλες συναρτήσεις υπάρχουν είδη για ποιο εύκολη χρήση και είναι μερικές φτιαγμένες για την χρήση διάφορων υλικών μέσων που υπάρχουν εξωτερικά από την πλακέτα **(Led, Buzzer, αισθητήρες,κινητήρες,κ.α…)**

Τώρα θα ξεκινήσουμε με μερικές συναρτήσεις του **Arduino** έτσι ώστε να τα καταλάβουμε και μετά θα κάνουμε μερικά μικρά παραδείγματα για το καθένα έτσι ώστε να είναι ποιο εύκολο για εμάς .



Εικόνα 5.1:(https://docs.arduino.cc/programming/)

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Ότι θέλουμε να μάθουμε σχετικά με τον κώδικα του **Arduino**,μπορούμε να μπούμε στον υπερσύνδεσμο του **official site** τους έτσι ώστε να μάθουμε ότι χρειάζεται για ξεκινήσουμε με την βάση ή την εξέλιξη ενός πρότζεκτ μας

Ας αρχίσουμε με της βασικές συναρτήσεις που υπάρχουν,πρώτα με της ψηφιακές εισόδου και εξόδου, **DIGITAL I/O** όπου λειτουργούν σαν **input ή output** τα Pins και έχουν λογικές τιμές **High** ή **Low**

Αυτές οι συναρτήσεις είναι οι εντολές : **digitalRead(), digitalWrite(), pinmode()**

digitalRead(pin):Διαβάζει την τρέχουσα λογική κατάσταση τιμής **Low & High.**Απο ένα **Pin** που έχει δηλωθεί ως **Input**

```
int buttonState = digitalRead(2);  // Διαβάζει την είσοδο από το pin 2
```

Εικόνα5.2:Παράδειγμα digitalRead()

digitalWrite(pin,mode):Στέλνει μία ψηφιακή τιμή (**High,Low**),σε ένα **Pin** που εχει δηλωθεί ως **Output**

```
digitalWrite(13, HIGH);   // Στέλνει ρεύμα στο pin 13
digitalWrite(13, LOW);    // Κλείνει το ρεύμα
```

Εικονα5.3:Παράδειγμα digitalWrite()

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

pinMode(pin,mode):Καθορίζει κάποιο από τα **Pin** να λειτουργήσουν σαν **είσοδος ή έξοδος**

```
pinMode(13, OUTPUT);  // Το pin 13 γίνεται έξοδος
pinMode(2, INPUT);    // Το pin 2 γίνεται είσοδος
```

Εικονα5.4:Παράδειγμα pinMode()

INPUT:Το **Pin** διαβάζει σήματα π.χ(Ένα κουμπί)

OUTPUT:Το **Pin** στέλνει ένα σήμα π.χ(Ένα LED)

INPUT_PULLUP:Το **Pin** εισόδου έχει εσωτερική αντίσταση προς **+5V**(για κουμπιά χωρίς εξωτερική αντίσταση)

Τώρα θα κάνουμε ένα παράδειγμα και με τις τρείς συναρτήσεις σε ένα πρόγραμμα έτσι ώστε να τα καταλάβουμε καλύτερα ,χρησιμοποιώντας φυσικά την πλατφόρμα **Wokwi**, για να είναι ποιο εύκολο για εμάς και θα χρησιμοποιήσουμε μόνο ένα εξάρτημα το όποιο θα είναι ένα απλό **Button** και ένα **Led:**

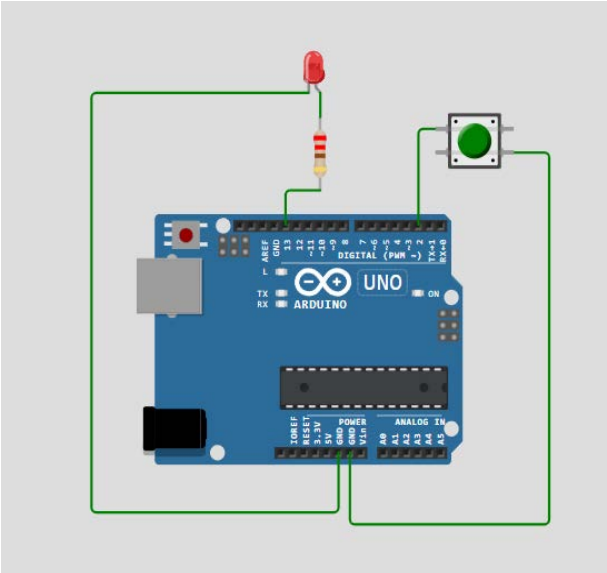
```
void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}

void loop() {
  int buttonState = digitalRead(2);

  if (buttonState == LOW) {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
}
```

Εικονα5.5(Παράδειγμα και με τις 3 συναρτήσει ς

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156



Εικονα5.6(Κύκλωμα με Led & Button)

Στο παράδειγμα που έχουμε κάνει όταν θα πατάμε το **Button** θα ανάβει το **Led.** Οπότε το κύκλωμα αυτό χρειαζόταν και τις τρεις συναρτήσεις που είχαμε,αναγνωρίζουμε ποια **Pin** είναι για **output** & **input** διαβάζουμε την **είσοδο** του **button** και για **έξοδο** το **Led**

Τώρα θα δούμε τις αναλογικές συναρτήσεις η αλλιώς **AnalogI/O,**
analogRead()
analogReadResolution()
analogReference()
analogWrite()
analogWriteResolution()

analogRead(pin): Διαβάζει μια αναλογική είσοδο από **A0** μέχρι **A5** και παίρνει τιμές από 0 μέχρι το 1023

```
int value = analogRead(A0);  // Διαβάζει αναλογική τιμή από A0
```

Εικονα5.7(Παράδειγμα analogRead())

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

analogReadResolution(bits): Η συνάρτηση ουσιαστικά ρυθμίζει την ανάλυση των **bit** του **analogRead** όπου αυτό έχει αυτόματα **10bit** μπορούμε να το κάνουμε **12bit** ,**14** ή ακόμη και **16 bit** και φυσικά το κάθε ένα θα έχει μέχρι άλλο όριο τιμών που θα μας δίνουν

```
analogReadResolution(12);   // Ανάλυση 12-bit → τιμές 0 έως 4095
int value = analogRead(A0);
```

Εικονα5.8: (Παράδειγμα analogReadResolution)

analogReference(type): Ορίζει το μέγιστο όριο τάσης από την αναφορά που παίρνουμε από το **analogRead(),** έχει διάφορα **types** όπως **INTERNAL** και **EXTERNAL**,αλλά είναι ανάλογα την πλακέτα μας και το είδος της,λίγο ποιο κάτω θα δώσουμε διάφορα παραδείγματα και θα έχουμε εικόνες με όλο το υλικό ανάλογα την πλακέτα και το **version** της

```
analogReference(INTERNAL);  // Ορίζει ως μέγιστο τα 1.1V
```

Εικονα5.9: (Παράδειγμα analogReference())

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

Arduino AVR Boards (UNO, Mega, Leonardo, etc.)

- **DEFAULT:** the default analog reference of 5 volts (on 5 VDC Arduino boards) or 3.3 volts (on 3.3 VDC Arduino boards).
- **INTERNAL:** a built-in reference, equal to 1.1 volts on the ATmega168 or ATmega328P and 2.56 volts on the ATmega32U4 and ATmega8 (not available on the Arduino Mega).
- **INTERNAL1V1:** a built-in 1.1 VDC reference (Arduino Mega only).
- **INTERNAL2V56:** a built-in 2.56 VDC reference (Arduino Mega only).
- **EXTERNAL:** the voltage applied to the AREF pin (0 to 5 VDC only) is used as the reference.

Arduino Renesas Boards (UNO R4, Portenta C33)

- **AR_DEFAULT:** the default analog reference of 5 volts.
- **AR_INTERNAL:** a built-in reference, equal to 1.5 Volts on the RA4M1 of the UNO R4.
- **AR_INTERNAL_1_5V:** a built-in reference, equal to 1.5 VDC on the R7FA6M5 of the Portenta C33.
- **AR_INTERNAL_2_0V:** a built-in reference, equal to 2.0 VDC on the R7FA6M5 of the Portenta C33.
- **AR_INTERNAL_2_5V:** a built-in reference, equal to 2.5 VDC on the R7FA6M5 of the Portenta C33.
- **AR_EXTERNAL:** the voltage applied to the AREF pin (0 to 5 VDC only) is used as the reference.

Arduino SAMD Boards (Zero, etc.)

- **AR_DEFAULT:** the default analog reference of 3.3 VDC.
- **AR_INTERNAL:** a built-in 2.23 VDC reference.
- **AR_INTERNAL1V0:** a built-in 1.0 VDC reference.
- **AR_INTERNAL1V65:** a built-in 1.65 VDC reference
- **AR_INTERNAL2V23:** a built-in 2.23 VDC reference
- **AR_EXTERNAL:** the voltage applied to the AREF pin is used as the reference

Ουσιαστικά αυτό που χρειαζόμαστε είναι ανάλογα της προτιμήσεις μας, θα βρούμε και τα υπόλοιπα παραδείγματα στην αρχική σελίδα του **Arduino** κάτω από την συνάρτηση που έχουμε,στον ιστότιπο:https://docs.arduino.cc/language-reference/en/functions/analog-io/analogReference/

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

analogWrite(): Δίνει αναλογική τιμή(**PWN WAVE**) σε ένα **Pin**,μπορεί να χρησιμοποιηθεί για να δώσουμε λιγότερη τάση σε ένα **Led** η ένα **Moter,** δέχεται τιμές από 0 μέχρι 255 όπου το 0 είναι **Off** και το 255 On, δεν μπορεί να πραγματοποιηθεί σε κάθε **Pin** είναι ανάλογα το **Board** μας

```
analogWrite(9, 128);  // Μισή φωτεινότητα στο LED στο pin 9
```

Εικονα5.10(Παράδειγμα analogWrite()**)**

analogWriteResolution(): Ρυθμίζει την αναλογική τιμή που δίνουμε με την **analogWrite(),** επίσης πρέπει να προσέχουμε την συνάρτηση επειδή δεν υποστηρίζεται από όλες τις πλακέτες

```
analogWriteResolution(10);  // Τώρα η analogWrite() δέχεται  απο 0-1023
analogWrite(9, 900);
```

Εικονα5.11:Παράδειγμα analogWriteResolution

Τέλος θα αναφέρουμε τις συναρτήσεις **Advanced I/O:noTone(),pulsein(),pulseinLong(),shiftin(),shiftIn(),shiftOut(),tone()**

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

tone(pin,frequency,duration):Κάνει ρυθμίσει την παραγωγή συχνότητας και ήχου σε ένα **Pin** και με ότι διάρκεια θέλουμε, είναι χρήσιμο σε **buzzer** και ηχεία

```
tone(10, 800);       // Ήχος 800Hz στο pin 10 (μέχρι να τον σταματήσεις)
tone(10, 400, 200);   // Ήχος 400Hz για 200ms
```

Εικονα5.12: Παράδειγμα tone

noTone(pin): Σταματάει τον ήχο του συγκεκριμένου **Pin**

```
noTone(8);  // Σταματάει τον ήχο στο pin 8
```

Εικονα5.13: Παράδειγμα noTone

pulseIn(pin,value): Μετράει το πόσο διαρκεί ένας παλμός σε ένα **Pin(High ή Low)**

```
long duration = pulseIn(7, HIGH);  // Πόση ώρα ήταν HIGH το pin 7
```

Εικονα5.13: Παράδειγμα pulseIn

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

pulseInLong(pin,value): Κάνει το ίδιο με το **pulseIn** αλλά για μεγαλύτερους χρόνους

shiftOut(dataPin,clockPin,bitOrder, value): Στέλνει τα δεδομένα σειριακά σε άλλο **chip .** Πρώτα, θα δούμε την κάθε περίμετρο για να καταλάβουμε μετά ποιο εύκολο το παράδειγμα μας

1.**dataPin:** Το **Pin** που στέλνουμε τα **bit** ένα μετά το άλλο (1 ή 0)

2.**clockPin:** Το Pin που στέλνει τον χρονισμό και πότε θα στείλει κάθε **bit,** από **(HIGH →LOW & LOW→HIGH)** στέλνει ένα **bit** από το **dataPin.**

3.**bitOrder:** Η σειρά που στέλνονται τα **bit,** με αναλογία την σημαντικότητα τους,**MSBFIRST(πιο σημαντικό bit) & LSBFIRST(λιγότερο σημαντικό bit)**

4.**value:** Το δεδομένο **byte** που θέλουμε να στείλουμε,περιέχει 8 bit

shiftIn(dataPin,clockPin,bitOrder): Κάνει το αντίθετο του **shiftOut,** διαβάζει τα δεδομένα σε σειριακή μορφή από εξωτερική μορφή συσκευής .

Τώρα που είδαμε και το κώδικα, μπορούμε να καταλάβουμε ότι δεν είναι δύσκολο να αρχίσουμε με ένα πρότζεκτ,πρέπει πρώτα να σκεφτόμαστε τι πρότζεκτ πρέπει να κάνουμε μετά θα δούμε τις επιλογές στην πλακέτα(**Mega,Uno)** που θα κάνουμε το κύκλωμα μας,τις δυνατότητες της ανάλογα τα εξαρτήματα μας και στο τέλος να δούμε το πως μπορούμε να το εξελίξουμε σε ένα μεγαλύτερο σημείο,επίσης πολλές από τις συναρτήσεις που μάθαμε μπορούνε να καταφέρουν την επικοινωνία παραπάνω και προς εξωτερικά **chips** αλλά παράλληλα και να προσθέσουμε επιπλέον επικοινωνία και ασύρματα ή με εφαρμογές κινητών.Άρα να εξελίξουμε κάτι με σύνθετο τρόπο,είναι εύκολο και επειδή όλα τα δεδομένα είναι **Open source,** μπορούμε να πάρουμε ιδέες ή να καταφέρουμε παραπάνω από ότι γνωρίζαμε αρχικά

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

ΒΙΒΛΙΟΓΡΑΦΙΑ

Wokwi:

https://wokwi.com/

https://wokwi.com/projects/429504401538584577

https://wokwi.com/projects/429552192844012545

https://wokwi.com/projects/431018882580302849

Tinkercard[4]: Σελιδα(16-17)
https://www.tinkercad.com/

https://www.tinkercad.com/learn/circuits?collectionId=O0K87SQL1W5N4P2

SimulIDE[5]: Σελιδα(19-20)

https://simulide.com/p/

Arduino[1](8-12):

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

https://en.wikipedia.org/wiki/Arduino#Official_boards

https://en.wikipedia.org/wiki/Arduino#Sketch

https://en.wikipedia.org/wiki/Arduino#Libraries

https://docs.arduino.cc/programming/

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156

https://docs.arduino.cc/language-reference/

https://docs.arduino.cc/hardware/

https://docs.arduino.cc/learn/

Institutional Repository - Library & Information Centre - University of Thessaly
02/09/2026 09:13:48 EEST - 216.73.216.156