



Φύλλο εργασίας στον Προγραμματισμό με Arduino

Διδακτικές Ενότητες: Βρόχοι 'Ελέγχου' και 'Επανάληψης'

Όνοματεπώνυμο 1: _____

Όνοματεπώνυμο 2: _____

Πατήστε στο δεσμό [TINKERCAD](#) και επιλέξτε την εντολή

Sign in

Sign in with Google

Στη συνέχεια για να εργαστείτε στην πλατφόρμα επιλέξτε

Δώστε το όνομα χρήστη και τον κωδικό του Gmail σας για να μπειτε στην πλατφόρμα.

Μετά την είσοδό σας επιλέξτε

Circuits

και

Create new Circuit

για να μεταβείτε στο πεδίο του εξομοιωτή Arduino.

Τρόπος εργασίας / στάδιο:

Δημιουργείτε το κύκλωμα όπως φαίνεται στη φωτογραφία σε κάθε στο κάθε στάδιο. Αφού τελειώσετε το κύκλωμα θα συζητήσουμε για τη δομή του κώδικα και τη σημασία του κάθε βήματος (ή μπλοκ). Έπειτα θα έχετε χρόνο για να απαντήσετε στις ερωτήσεις. Στη λήξη του χρόνου θα συζητάμε στην ολομέλεια για να δώσουμε όλοι μαζί μια λύση.

Παρακαλώ μην προχωράτε σε επόμενο στάδιο αν δεν έχουμε όλοι μας ολοκληρώσει το προηγούμενο.



Προσοχή: όταν δημιουργούμε κώδικα με μπλοκ, αυτομάτως δημιουργείται και το κείμενο σε C++. Όχι όμως το αντίστροφο!

Άρα είτε εργαζόμαστε σε: 'Blocks' ή 'Blocks + Text'
είτε μόνο σε: 'Text'

Are you sure?

Enabling the blocks editor will clear any code you have in the text editor. Are you sure you want to continue?

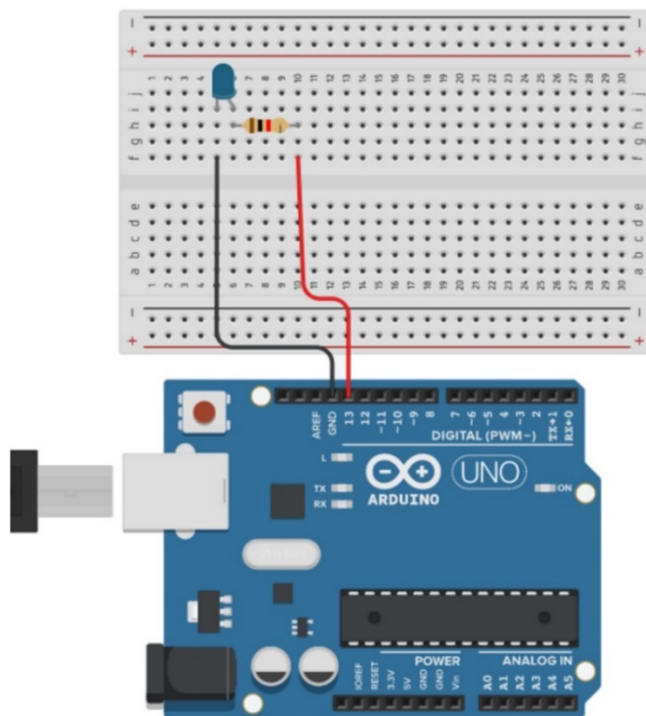
Continue

Cancel

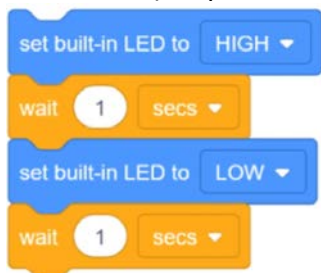
1^ο στάδιο: Δημιουργία του κυκλώματος Blink

Δημιουργήστε το κύκλωμα, όπως ακριβώς το βλέπετε στην επόμενη εικόνα και πατήστε

▶ Start Simulation



Κώδικας σε μπλοκ



Κώδικας C++

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

Αλλάξτε τα μπλοκ  σε  (όμοια και το αντίστοιχο με LOW) και προσέξτε τις αλλαγές και στον κώδικα.

Άσκηση: (Σημειώστε την αλλαγή μόνο στον κώδικα (μπλοκ & κώδικας C++)):

1. Από τον κώδικα σε C++ μορφή, μειώστε το χρόνο αναμονής σε ½ δευτερόλεπτο.	2. Στη συνέχεια αφήστε συνεχώς ανοικτό το φως του led.

3. Προσθέστε και ένα 2^ο led. Δηλώστε – το και στο κώδικα	
4. Αλλάξτε τον κώδικα ώστε να ανάβουν σε σειρά και να σβήνουν σε σειρά με αντίθετη ροή (ανάβει το 1^ο μετά το 2^ο) και (σβήνει το 2^ο και μετά το 1^ο)	

Παρακαλώ μην προχωράτε σε επόμενο στάδιο αν δεν έχουμε όλοι μας ολοκληρώσει το 1^ο στάδιο.

2^ο στάδιο: Δημιουργία του κυκλώματος Blink με κουμπί buzzer

Δημιουργήστε το κύκλωμα, όπως ακριβώς το βλέπετε στην επόμενη εικόνα και πατήστε

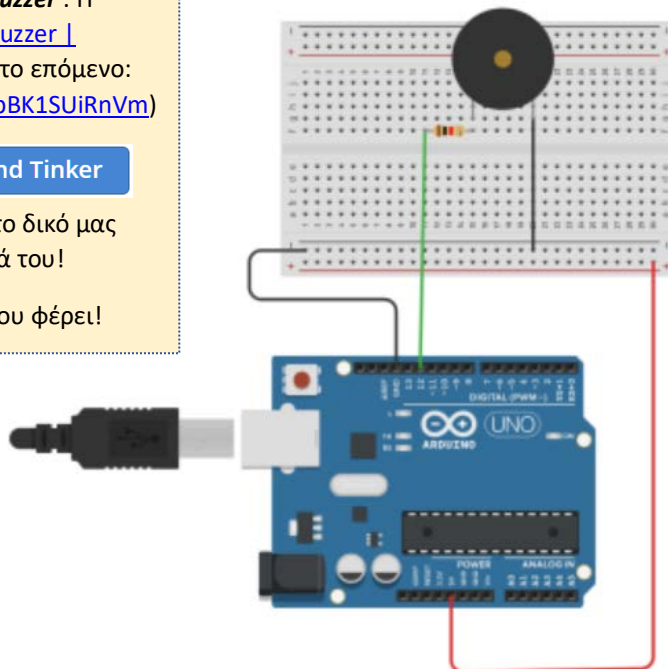
▶ Start Simulation

Μπορούμε να βρούμε το κύκλωμα κάνοντας μια αναζήτηση της φράσης: **'tinkercad buzzer'**. Η επιστροφή μας δίνει και το δεσμό ['Buzzer | Tinkercad'](#) (ή απλά πληκτρολογήστε το επόμενο: <https://www.tinkercad.com/things/bBK1SUiRnVm>)

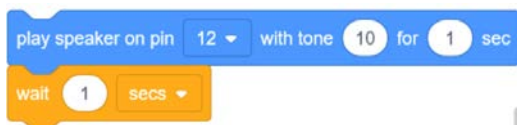
Copy and Tinker

Στη συνέχεια επιλέγουμε: για να αντιγράψουμε το κύκλωμα στο δικό μας λογαριασμό. Έτσι έχουμε τον κώδικά του!

ΠΡΟΣΟΧΗ: στη μορφή του κώδικα που φέρει!



Κώδικας σε μπλοκ



Κώδικας C++

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(12, OUTPUT);
6 }
7
8 void loop()
9 {
10  tone(12, 29, 1000); // play tone 10 (A#0 = 29 Hz)
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Άσκηση: (Σημειώστε την αλλαγή μόνο στον κώδικα (μπλοκ & κώδικας C++)):

Πώς θα μπορούσαμε να προσθέταμε ένα led με τέτοιο τρόπο ώστε όποτε ακούγεται ο ήχος του buzzer ν' ανάβει;

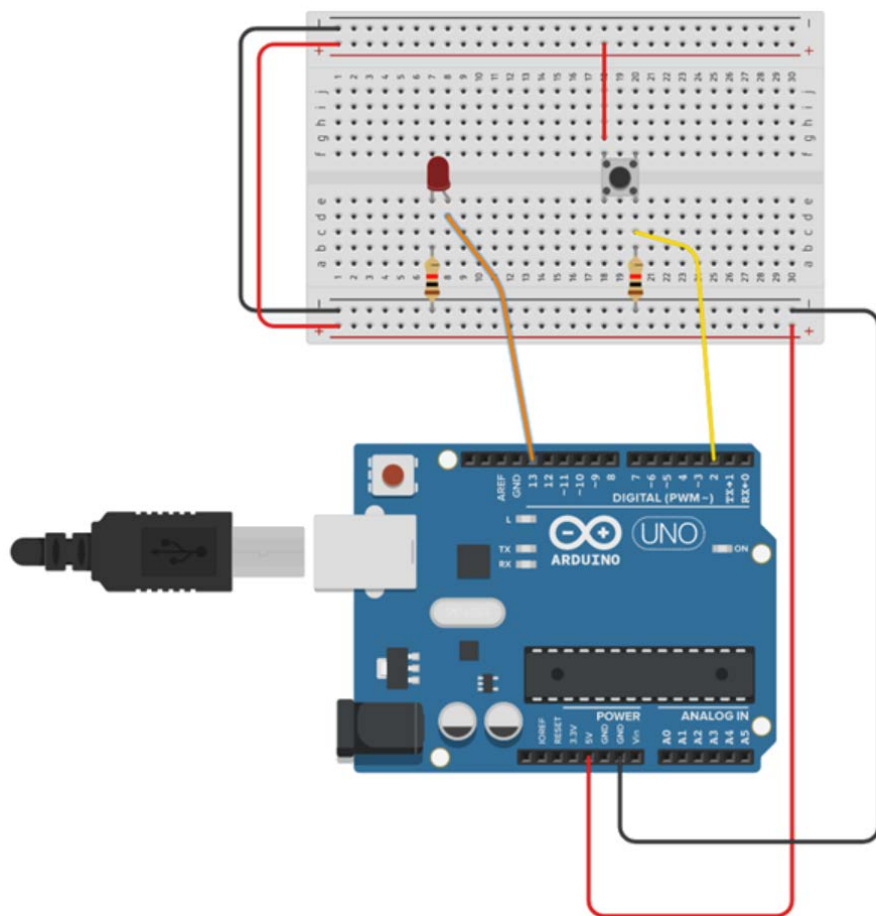
--	--

Παρακαλώ μην προχωράτε σε επόμενο στάδιο αν δεν έχουμε όλοι μας ολοκληρώσει το 2^ο στάδιο.

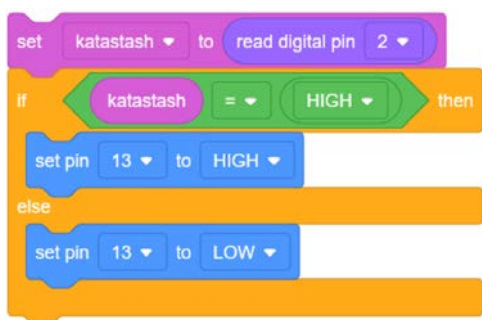
3^ο στάδιο: Δημιουργία του κυκλώματος Blink με κουμπί ελέγχου

Δημιουργήστε το κύκλωμα, όπως ακριβώς το βλέπετε στην επόμενη εικόνα και πατήστε

Start Simulation



Κώδικας σε μπλοκ



Κώδικας C++

```
1 // C++ code
2 //
3 int katakash=0;
4
5 void setup()
6 {
7   pinMode(2, INPUT);
8   pinMode(13, OUTPUT);
9 }
10
11 void loop()
12 {
13   katakash = digitalRead(2);
14   if (katakash == HIGH) {
15     digitalWrite(13, HIGH);
16   } else {
17     digitalWrite(13, LOW);
18   }
19   delay(10); // Delay
20 }
```

ΠΡΟΣΟΧΗ: Για να δημιουργήσουμε μια νέα μεταβλητή με μπλοκ:

Variables → Create variable. → και δηλώνουμε ένα όνομα, π.χ. **katakash**

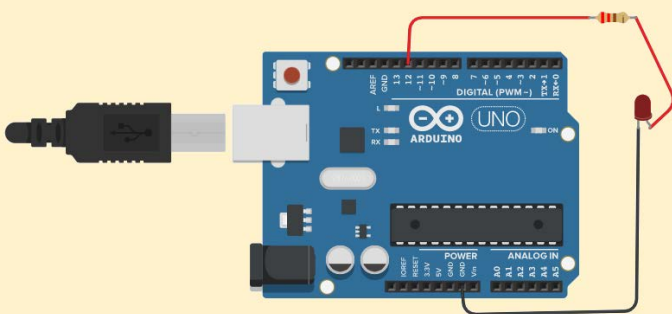
Θα εμφανιστούν 3 στοιχεία: Για να τη φωνάζουμε όποτε χρειαστεί Για να της ορίζουμε μια τιμή Για αλλαγή της τιμής

(Βλ. παράδειγμα στο τέλος)

Άσκηση: (Σημειώστε την αλλαγή στον κώδικα (μπλοκ – επιγραμματικά & κώδικας C++)):

1. Πώς θα μπορούσαμε να προσθέταμε ακόμα ένα led και να ανάβει ταυτόχρονα με το 1ο;	2. Και πώς πρέπει να φτιάξουμε τον κώδικα ώστε όταν πατάμε το κουμπί να ανάβει το 1 ^ο και όταν το αφήνουμε να ανάβει το 2 ^ο ;
3. Αφαιρέστε το επιπλέον led, διαγράψτε και τον κώδικά του. Πώς θα προσθέσετε ένα buzzer ώστε όταν πατάτε το κουμπί να ανάβει το led και να ακούγεται και το buzzer;	

Παράδειγμα δήλωσης και χρήσης σταθερών τιμών και μεταβλητών με κώδικα C++



```
//Δηλώνω τις σταθερές τιμές (constants)
//και τις μεταβλητές (variables)
//Δηλώνω τη μεταβλητή LED
int led;
//Δηλώνω τη σταθερά ταχύτητα Blink του LED
//Η τιμή της δεν αλλάζει πια!
const int pause = 1500;

void setup()
{
  // Εδώ ο κώδικας που θα τρέξει μια φορά μόνο:
  //Δίνω τιμή στη μεταβλητή LED
  led = 12;
  //Βάζω το LED pin ως Output
  pinMode(led, OUTPUT);
}

void loop()
{
  //Ότι κώδικα βάλω εδώ θα τρέχει συνέχεια:
  //Ανάβει και σβήνει το LED για pause χρόνο
  //στο τέλος μειώσαμε το χρόνο σε 250 ms
  digitalWrite(led, HIGH);
  delay(pause);
  digitalWrite(led, LOW);
  delay(pause - 125);
}
```