

מדריך ארדואינו

תוכן עניינים

2	מבוא ל-IOT ולארדואינו
4	הדפסות ל-console
9	נורה מהבהבת על ה-Board
11	רכיב הלייזר
14	חיישן קולט לייזר
18	קודן keypad
23	זמזום
25	Blinking Led - נורה מהבהבת
27	מונה זמן
30	אינפרא אדום
33	נורה מחליפה צבעים
36	חיישן מודד מרחק
39	חיישן מודד דציבלים

מבוא לז'אנר ולארדואינו

העתיד טומן בחובו כל כך הרבה סימני שאלה, רעיונות והמצאות שאנו לא מעלים על דעתנו.

לפי כ-20 שנה, בשנת 1999, עלתה פרסומת של חברת סלקום שהציגה את העתיד - טלפונים חכמים, תקשורת מהארץ לחו"ל בשיחת וידאו, אפשרויות בלתי נגמרות והכל כאן ועכשיו ומידי. אנשים לא האמינו לכך. לא האמינו שאכן כך יהיה... ואם יהיה - אז לבטח לא ב-2 עשורים... ההתפתחות העצומה הזו תוקפת אותנו מכל הכיוונים והיא בעיקר מכוונת לעולם של AI ו-IoT.

במדריך זה אסקר אפשרויות אלקטרוניקה ותכנות, יצירת מערכות קטנות לבית חכם והכל באמצעות רכיבי הארדואינו וכתובה ב.C++
עליכם להצטייד ברכיבים מראש, לרכוש ב-EBAY, להוריד את סביבת הפיתוח של ארדואינו ולהתחיל לעבוד.

הצטרפו אלינו למדריך ה-IoT שלנו!

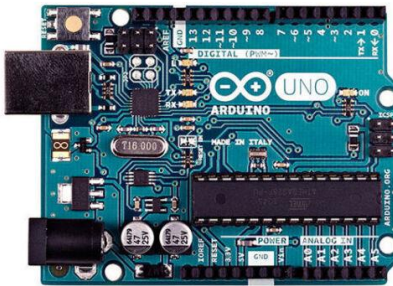
Internet
Of
Things

הוכיחו לעצמכם שגם אתם יכולים לתכנת מערכות בית חכם לגמרי בעצמכם!
על מנת להגיע מוכנים לתחילת העבודה עם ארדואינו במדריך זה, ודאו כי:

מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©

[1. התקינו את סביבת הפיתוח עם ארדואינו](#)

[2. ריכשו מ EBAY את ה BOARD של ארדואינו UNO](#)



הדפסות לconsole

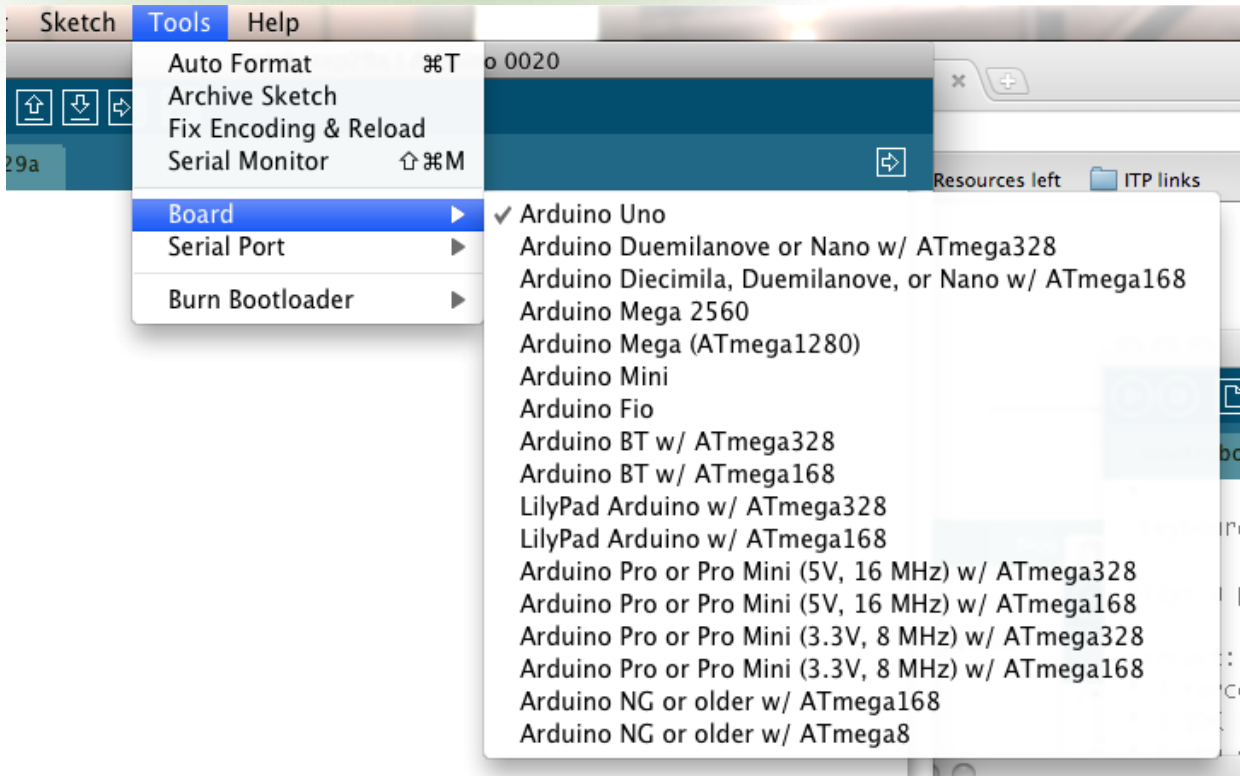
לאחר שרכשתם את הBOARD והתקנתם את סביבת הפיתוח של ארדואינו (כיום ניתן גם לוותר על ההתקנה ולתכנת באמצעות Online Editor חדשני, לנוחיותכם) נוכל להתחיל להעניק פקודות והוראות בסיסיות לארדואינו שלנו.

הפקודות הבסיסיות שנלמד בשיעור הזה הם הדפסות ישירות לConsole מובנה מראש בתוך התוכנה. לא נתחיל ישירות מהענקת פקודות לרכיבים חיצוניים, קרי סנסורים, נורות וכדומה...

המדריך נבנה בצורה איטית וסבלנית ומומלץ להישאר שלב אחר שלב ולא לקפוץ קדימה. כיאה לשפות תכנות רבות, נרצה גם במקרה של ארדואינו להפעיל הדפסות Debug.Log שרק אנחנו נוכל לראות (המתכנתים).

על מנת להפעיל קונסולה ולהדפיס פקודות לוגיות בסיסיות (משתנים, תנאים ולולאות) עלינו לבצע מספר פעולות קודמות:

1. עלינו לעבור על [מדריך הלוגיקה](#) באתר עולם ההייטק ולוודא כי אנו סגורים על מושגים בסיסיים בתכנות וכן כיצד נוכל ליצור משתנים, מהם האופרטורים הבסיסיים, לכתוב פונקציות, מהו תנאי פשוט ומה תפקידה של לולאה.
2. עלינו לחבר את הBOARD שלנו עם כבל USB למחשב ולפתוח את סביבת הפיתוח. בסביבת הפיתוח, עלינו ללחוץ על 'Tools' בסרגל העליון ולהגדיר כי אנו משתמשים ב-Arduino Uno.



3. עלינו לוודא כי באפשרות 'PORT' אנו רואים את הCOM4 או הCOM3 (כל מחשב מזהה לפי הBOARD שנרכש מEBAY).

4. לסיום, נלחץ על 'מסך סיריאל' באותו תפריט Tools. ניתן לפתוח את מסך הקונסולה הזה גם דרך שילוב המקשים Ctrl+Shift+M, לנוחיותכם.

כעת, כאשר מסך הסיריאל פתוח לפנינו - נוכל לראות כי ישנן שתי פונקציות עיקריות:

האחת – `void setup()` אשר מה שנכתוב בתוכה יקרה תמיד פעם אחת בלבד, בעת התחלת העבודה עם הBOARD (למשל, בהרצה של התוכנה או בלחיצה על כפתור הRESET העגול האדום שיש על הBOARD).

השניה – `void loop()` אשר מה שנכתוב בתוכה יקרה בכל מאית שניה כאשר התוכנה רצה. מעין לולאה אינסופית שכזו... ללא הפסקה. לנצח. תוכלו לחשוב למשל מה נרצה שיקרה ללא הפסקה שוב ושוב? בדיקה מסוימת אולי על הבית החכם שלנו? אולי שורה שתרוץ שוב ושוב... ללא הפסקה?

כעת - נוכל לכתוב משתנים גלובליים (שיהיו מחוץ לשתי הפונקציות וששתיהן יכירו אותה). נוכל לבצע מניפולציות על המשתנים שיצרנו. למשל, משתנה גלובלי יהיה מספר - המניפולציה בתוך `setup` תהיה המספר כפול עצמו ואז נרצה גם להדפיס! (ראה ערך: קלט - מניפולציה - פלט) במדריך הלוגיקה שלנו.

כאשר נרצה להדפיס, נצטרך תחילה לכתוב בתוך פונקציית `setup` שלנו שורה שקוראת להתחלת העבודה עם הקונסול. `Serial.begin(9600)` נשים לב כי בתחתית מסך הסריאיל באמת כתוב לנו שהשידור הוא 9600! כך אנו מתחברים אליו.

כמו גם, נוכל ליצר בתוך הפונקציות אפשרות של הדפסה באמצעות שורת הקוד `Serial.println()` וכפרמטר נכתוב את המשתנה שנרצה להציג או סטרינג כלשהו בגרשיים. הביטוי בסרטון המצורף.

כלים

1. בהתבסס על הצפייה בסרטון ומדריך הלוגיקה, צרו 3 משתנים גלובליים מסוג `int` והדפיסו ב `setup` את הממוצע שלהם.
2. צרו משתנה מסוג `int` והציגו את המספר כפול עצמו ב `setup`.
3. צרו משתנה מסוג `int` והציגו ב `setup` האם המספר זוגי או אי זוגי.
4. צרו משתנה מסוג `int` והציגו ב `loop` לולאה אשר מציגה את כל המספרים מ 1 ועד המספר שנקלט. האם יש סוף ללולאה הזו?
5. אתגר: חפשו ברשת - קילטו משתנה מסוג `String` והערך שלו זה שם של מדינה בעולם. הציגו ב `setup` את כמות האותיות של המדינה (רמז: `length`).

מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©



נורה מהבהבת על הBoard

בלוח הארדואינו (להלן: 'Board') יש לנו אפשרויות של משחק עם דברים מוכנים מראש - BuiltIn. למשל, אנו רואים כי יש נורה צהובה ליד הפינים של אזור הDigital שלנו... טרם העבודה עם רכיבים חיצוניים (שיקרו ממש בשיעור הבא ועליכם יהיה לרכוש אותם מEBAY) עדיין נתמקד במבוא לארדואינו וננסה לגרום לנורה שלנו להידלק ולהיכבות כל שניה בLOOP אינסופי לנצח.

ראשית, נכיר פונקציה בסיסית של התחברות לרכיבים - `pinMode()`;

שמה ואנו מכניסים ב2 פרמטרים פנימה את (1) שם הרכיב ו(2) האם הוא קולט משהו (INPUT) או פולט משהו החוצה (OUTPUT).

במקרה שלנו, שם הנורה הצהובה הוא LED_BUILTIN ואנו קולטים משהו או פולטים משהו לדעתכם? (נזכיר, רוצים שהנורה תהבהב). על מנת לחשוף את הפיתרון של הפונקציה המלאה אותה יש לשים בsetup -

```
pinMode(LED_BUILTIN , OUTPUT);
```

לסיום, בתוך פונקציית הLoop נרצה לייצר מצב בו לעיתים הנורה תהיה דולקת (נקרא גם: HIGH) או שתהיה כבויה (נקרא גם: LOW). את האפשרות הזו נכתוב בתוך פונקציה בשם digitalWrite (המקבילה שלה digitalRead אם היה מדובר בINPUT ולא OUTPUT) אשר

מקבלת 2 פרמטרים. יכולים לחשוב איזה? וביניהם לייצר DELAY קל של שניה (1000 מייצגת שניה).

נסו בעצמיכם! הפיתרון המלא פה:

```
void Loop(){  
  
    digitalWrite(LED_BUILTIN , HIGH);  
    delay(1000);  
    digitalWrite(LED_BUILTIN , LOW);  
    delay(1000);  
  
}
```

הערה חשובה: לאחר כתיבת כל קוד נלחץ על כפתור הקומפליציה) בצורת V ועל כפתור ה UPLOAD כמובן (בצורת חץ). שניהם נמצאים במסך כתיבת הקוד עצמו (ממש מעל הקוד ומתחת לתפריט העליון).



רכיב הלייזר

בשיעור זה נלמד לחבר רכיב חיצוני - לייזר ונגרום לו להידלק ולהיכבות בדיוק כמו שביצענו בשיעור הקודם על נורת הדיגיטל הצהובה הבuiltin.

ראשית, עלינו לרכוש את רכיב הלייזר מEBAY מומלץ לרכוש אותו ישירות עם חישן קולט הלייזר שעליו נלמד בשיעור הבא. לינק לרכישה נמצא [בלחיצה כאן](#)



נשים לב כי לרכיב הלייזר שלנו יש 3 חיבורים. באמצעות ג'אמפרים) [חוטים](#)
[ארוכים לרכישה כאן](#) (נוכל לחבר אותו פנימה לתוך ה BOARD שלנו בשלושת
המקומות הבאים:

את החוט בצד הימני ביותר של הלייזר נחבר ל GND (פין שנמצא בצד המקביל
ל digital, באזור ה power). משמעות פין זה היא GROUND והוא שמשמש
כמקום בו מפלס החשמל הוא 0 וולט.

את החוט האמצעי נשאיר ללא חיבור כרגע.

את החוט הצד השמאלי ביותר של הלייזר נחבר לפין כלשהו על לוח ה BOARD
שלנו. למשל, פין מספר 10.

כעת, יש ללייזר שלנו גם חיבור חשמל וגם חיבור לאחד מהפינים הדיגיטליים. נוכל לתקשר איתו!
במקום להגיד ל led_builtin שיהיה ב OUTPUT, נרצה להגיד ל Laser שלנו וכמו גם במקום להגיד
ל led_builtin שיבצע HIGH ו LOW בהתאמה, נגיד ל Laser שלנו. העניין שאם נכתוב ישירות
במקום ה LED את ה LASER, נקבל שגיאת קומפליציה... הוא לא מכיר את LASER. עלינו לייצר
משתנה בשם זה ולהגדיר לו שהוא 10! פשוט כך. יש לכם רעיון איך לכתוב זאת? הפיתרון בעמוד
הבא.

```
int laser = 10;

void setup(){

    pinMode(laser, OUTPUT);
}

void loop(){

    digitalWrite(laser , HIGH);
    delay(1000);
    digitalWrite(laser , LOW);
    delay(1000);
}
```

חיישן קולט לייזר

אנחנו מתקדמים בקצב מעולה וכעת, נרצה לקבל מידע (לקלוט - INPUT) מתוך רכיב חיישן קולט הלייזר שלנו - האם הלייזר פוגע בו או לא?

ראשית, נחבר את הרכיב שלנו ללוח ה-BOARD בשלושת חוטיו!

1. חוט שמאלי ביותר ל-GND (שימו לב יש גם בצד ה-DIGITAL שלנו)
2. חוט אמצעי למספר מסוים. קרי 12.
3. חוט שמאלי ל-V5 (נמצא בצד של ה-Power).

כעת, צפו בסרטון והיווכחו בעצמכם - הקולט לייזר מחזיר מידע 0 או 1 באמצעות פונקציה שנקראת `digitalRead(sensor);`

נסו לשים פונקציה זו בתוך Serial שלנו (כמובן לא לשכוח `begin` ו-`pinMode` לסנסור) והיווכחו בעצמכם!

```
int sensor = 12;

void setup() {

    // put your setup code here, to run once:
    pinMode(sensor , INPUT);
    Serial.begin(9600);
    Serial.println('Start...');

}

void loop() {

    // put your main code here, to run repeatedly:
    Serial.println(digitalRead(sensor));

}
```

שאלה: מהו הערך שיופיע בלולאה אינסופית? 0 או 1?

כמו גם, נרצה להמשיך את העבודה על הלייזר. מה יקרה כאשר הלייזר יפגע בחיישן? האם ה-0 משתנה ל-1 ב-SERIAL שלנו?...

במידה וכן - נסו ליצר בדיקה... if פשוט. אם זה 0 תעשה משהו, אם 1 תעשה משהו אחר...

מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©

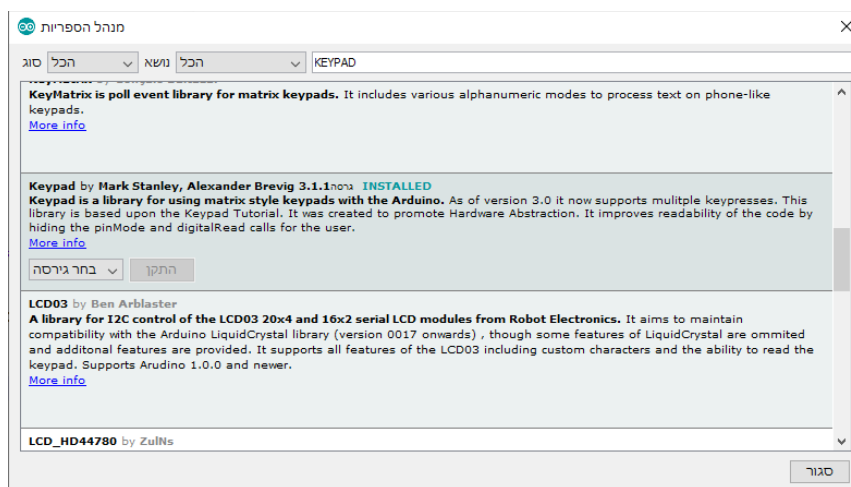


קודן keypad

בשיעור זה נעלה ברמה ונלמד כיצד לחבר רכיב חיצוני מסוג קודן ובמידה והקשנו סיסמא נכונה - נגרום לו לבטל את הלייזר הדולק שלנו!
ראשית, עלינו לרכוש את הקודן הפשוט ב EBAY שעולה \$1! [ליחצו כאן](#).

האינטרנט מלא במדריכים בנושא keypad הבסיסי ואנו נישען על קוד כתוב מראש ונטמיע אותו בקוד שלנו.

ראשית, בתוך הסרגל העליון נלחץ על 'סקיצה' ובפנים נצטרך לייבא ספרייה חדשה באמצעות 'ניהול הספריות'. (בקיצור מקשים +Shift+Ctrl לנוחיותכם) ראו בסרטון המצורף. עלינו לייבא את keypad שיצרו Mark Stanley ו Alexander Brevig –



בשיטוט מהיר בגוגל, נוכל לקבל קוד המכיל מערך של כלל התווים. נטמיע את הקוד הבא בקוד שלנו:

```
#include 'Keypad.h'
const byte ROWS = 4; // four rows
const byte COLS = 3; // three columns
char keys[ROWS][COLS] =
{
  {'1','2','3' },
  {'4','5','6' },
  {'7','8','9' },
  {'*','0','#' }
};
byte rowPins[ROWS] = {5, 6, 7, 8};
byte colPins[COLS] = {2, 3, 4};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
char KEY[4] = {'1','2','3','4'}; // default secret key
char attempt[4] = {0,0,0,0};
int z=0;
void setup()
{
  Serial.begin(9600);
}
void correctKEY() // do this if the correct KEY is entered
{
  Serial.println(' KEY ACCEPTED...!');
}
void incorrectKEY() // do this if an incorrect KEY is entered
{
  Serial.println('KEY REJECTED!');
}
void checkKEY()
{

```

```
int correct=0;
int i;
for ( i = 0; i < 4 ; i++ )
{
    if (attempt[i]==KEY[i])
    {
        correct++;
    }
}
if (correct==4)
{
    correctKEY();
}
else
{
    incorrectKEY();
    for (int zz=0; zz<4; zz++) // clear previous key input
    {
        attempt[zz]=0;
    }
}
void readKeypad()
{
    char key = keypad.getKey();
    if (key != NO_KEY)
    {
        switch(key)
        {
            case '*':
                z=0;
                break;
            case '#':
                delay(100); // added debounce
                checkKEY();
                break;
        }
    }
}
```

```
        default:
            attempt[z]=key;
            z++;
        }
    }
}
void loop()
{
    readKeypad();
}
```

כמו גם, עלינו לחבר את החוטים לפיני הDIGITAL שלנו על פי התמונה והסבר החיבורים החיוני -



לסיום, נוכל לראות כי בשורה מספר 14 כתוב לנו

```
char KEY[4] = {'1','2','3','4'}; // default secret key
```

ומשמעות הדבר היא שנוכל לשנות את תבנית המערך ובהקשה על מקש ה'#' נקבל תשובה האם הקוד שהזנו תקין או לא. בדיקת הקוד בתקינותה נעשית בפונקציה `checkKEY()`

שתי משימות לפניכם:

1. הוסיפו שורת פלט ל-Console שלנו על איזה מקש אנו לוחצים. יש לכם רעיון איפה נוכל לשלב פקודה זו? (רמז: נסו את הפונקציה של `void readKeypad()`).
2. נסו לגרום ללייזר להיכבות בעת הקשה על הקוד הנכון.



זמזם

בשיעור זה נלמד אודות הפעלת זמזם. הפעלת הזמזם דורשת מאיתנו ראשית, לרכוש את הזמזם מ EBAY מלחיצה על הכתובת [הבאה](#).

- כאשר הזמזם יהיה בידינו, נצטרך לחבר אותו באמצעות 3 ג'מפרים ל board שלנו.
- GND זהו הג'אמפר הראשון של הזמזם ויתחבר לאיזור ה GND ב POWER שלנו
- I/O זהו הג'אמפר השני שלנו ויתחבר לאזור ה DIGITAL לאחד מהפינים שאיתו נדבר.
- VCC זהו הג'אמפר השלישי שלנו ויתחבר ל V5 לקבל הספק כוח.

כעת, באמצעות קטע קוד בסביבת הפיתוח, נקשר את הזמזם ל pinMode:

```
int buzzer = 9; //buzzer to arduino pin 9

void setup(){

pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output

}
```

ולאחר מכן, נגרום לו להשמיע קול בתוך פונקציית ה loop באמצעות פונקציית tone. tone זו פונקציה שמייצרת גלי תדר על פין כלשהו. באמצעות פונקציה זו ניתן להפיק רק צליל אחד בכל פעם.

```
void Loop(){  
  
    tone(buzzer, 1000); // Send 1KHz sound signal...  
    delay(1000); // ...for 1 sec  
    noTone(buzzer); // Stop sound...  
    delay(1000); // ...for 1sec  
  
}
```

לסיום, כשהזמזם משמיע לנו קול - נסו לחשוב על לוגיקה פשוטה עם אחד מהסנסורים שכבר נלמדו. למשל, רק ברגע שחיישן קולט הלייזר, מקבל את הלייזר נוכל להפעיל את הזמזם. עבור אילו בעיות בקהילה שלנו, נוכל להתמודד באמצעות הזמזם והרכיבים שברשותכם? בהצלחה.



Blinking Led - נורה מהבהבת

אחד הרכיבים הבסיסיים ביותר בעולם ה-Arduino הוא הדלקה וכיבוי של נורות.

ראשית, נרכוש את הרכיב באמצעות הקישור [הבא](#)

כעת, נקח את הנורה שלנו ונראה כי יש לה שני חוטי מתכת היוצאות ממנה. את הארוך נחבר עם ג'אמפר בפין שנבחר ב-DIGITAL, בעוד את הקצר נחבר עם ג'אמפר ל-V5 באזור ה-POWER. ובאשר לקוד - מי שעובר איתי כבר דרך ארוכה ודאי כבר יודע כי עלינו לבצע משתנה גלובלי לנורה ולקשר אותה ב-setup למערכת שלנו.

```
int light = 2;
```

```
void setup(){  
  pinMode(light, OUTPUT);  
}
```

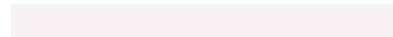
בחלק העיקרי - נסו לחשוב בעצמכם. כיצד נוכל לגרום לנורה להידלק ולהיכבות כל שניה?

הפיתרון בעמוד הבא אחרי הסרטון המצורף.



ולפיתרון שהובטח:

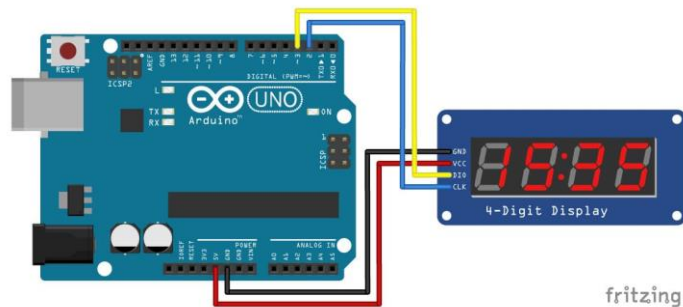
```
void loop(){  
  
    digitalWrite(2, HIGH)  
    delay(100); // ...for 1 sec  
    digitalWrite(2, LOW)  
    delay(100); // ...for 1sec  
  
}
```



מונה זמן

את המונה הזמן שלנו נוכל לרכוש בלחיצה כאן

את מונה הזמן שלנו נצטרך לחבר באמצעות ארבעה ג'אמפרים לGND, VCC ולשני פינים בזה אחר זה - בהתאמה.



כעת, נייבא ספרייה בשם TM1637.h מתוך הרשימה הקיימת ב-Arduino. (ראה ערך: שיעור הקודן. להבנה טובה יותר - הביטו בסרטון) נייצר 2 משתנים גלובליים האחד להצגת המספרים בסדר עולה (מ0...ועד 10) ושני להצגת המספרים בסדר יורד (מ10 ועד 0). כמו גם, לראשונה, נייצר משתנה אובייקט של משפחת TM1637Display המכיל את שני הפינים שהגדרנו. משתנה display של המופע שיצרנו יכיל את הקשר הישיר לרכיב שלנו ואיתו נדבר כעת (לא עם pins כמו שהיה עד כה) ראו הקוד:

```
#include <TM1637Display.h>
```

```
int top = 8;
```

```
int down = 9;
```

```
TM1637Display display(down, top);
```

בתוך setup נבצע את כל הפקודות שלנו. נרצה שכל ספירה תקרא פעם אחת ולכן נוותר על הloop.

קודם כל, נגדיר את הצבע האדום שיהיה בכל המספרים שירוצו.

```
display.setbrightness(0x0a)
```

בלי הגדרת הצבע, נישאר בצבע שחור ולא נראה את המספרים כלל.

לאחר מכן, נייצר לולאת for שתרוץ מהמספר 10 ועד המספר 0. מעין מונה שיספור

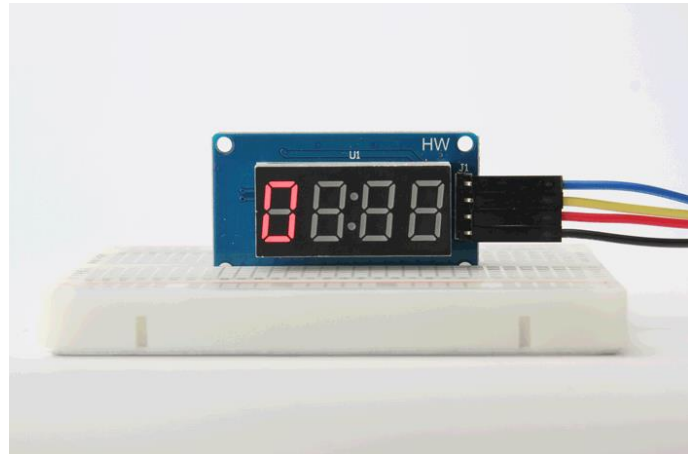
לאחור ... את התכנות נבצע באמצעות הפקודה `display.showNumberDec(i)`

ולא לשכוח לייצר `!delay`

נסו בעצמכם!

פיתרון פונקציית setup המלאה:

```
void setup(){  
  
  display.setbrightness(0x0a)  
  for (int i=10; i>=0; i--){  
    display.showNumberDec(i)  
    delay(1000);  
  }  
}
```



אינפרא אדום

בשיעור זה נלמד כיצד לגרום לחיישן אינפרא-אדום לשלוח גלי תדר לקולט שלו ואנחנו נדע האם הוא קלט את הגל או לא. תארו לעצמכם, כמו הסנסור של הלייזר והקולט שלו - כך גם פה. נוכל לשלוח גל מסוים, רק שהפעם אנחנו לא נראה את האינפרא. בניגוד ללייזר, ששם כן רואים.

ולצאת לדרך עם הרכיבים שלנו את הלינק לרכישה נוכל למצוא [כאן](#)

נחבר גם את האינפרא אדום שמשדר גלים וגם את הסנסור שקולט אותם ישירות לפינים הרלוונטים על הלוח שלנו וזה ההסבר המלא בהסתמך על התמונה הסמוכה:

IR Receiver קולט את המידע. נחבר אותו בהתאמה לפי שלושת הפינים - DATA לפין באזור הDIGITAL, VCC לפין הV5 באזור הGNDI POWER לפין הGND באזור הPOWER.

IR Transmitter משדר את המידע. נחבר אותו בהתאמה לפי שלושת הפינים - בדיוק כמו הקולט. - DATA לפין באזור הDIGITAL, VCC לפין הV5 באזור הGNDI POWER לפין הGND באזור הPOWER.

באשר לחלק התכנותי, נרצה לגרום לקולט האינפרא אדום להציג בSerial שלנו פלט כלשהו. 0 - כאשר אין אינפרא אדום שזז מולו וכל מספר אחר כאשר יש תנועה של האינפרא אדום מולו. תחילה, נחבר את הPins שלנו לתכנות באמצעות הפונקציה setup ולאחר מכן, בתוך פונקציית הloop (שרצה תמיד לנצח בלולאה אינסופית שכזו) נקבל גם את גלי התדר באמצעות פונקציית pulseIn וגם נגרום לTransmitter לשדר שלנו באמצעות פונקציית tone.

```
void setup(){
pinMode(7 , OUTPUT); //Transmitter
pinMode(8 , INPUT); //Receiver
Serial.begin(9600); //Start the serial console
}

void loop(){
Serial.println(pulseIn(8,LOW));
tone(7 , 100);
}
```

למעשה, פונקציית `pulseIn` מקבלת את הגלים ופונקציית `tone` משדרת אותם. נרצה שזה יקרה כל מאית שניה ולכן נשים את שתי השורות בתוך ה-`loop`.

באשר ל-`setup` ולהגדרות הבסיסיות של התוכנית שלנו, נכתוב בתוכן את השורות של קריאת הפינים הרלוונטים שאליהם חיברנו. אחד `OUTPUT` (שולח מידע) והשני `INPUT` (מקבל מידע) בהתאמה. *לא לשכוח לכתוב גם שורת קוד של `Serial.begin(9600)` שרק היא קובעת את האפשרות של התחלת ה-`Serial` (ראו ערך: שיעור מס' 2 - הדפסות ל-`Console`). במקום להציג מספרים כפלט על המסך, נוכל להפעיל זמזם כשיש תנועה של אינפרא-אדום...או להדליק נורה וכדומה. נסו לחשוב על פתרונות לקהילה שלכם שניתן ליישם באמצעות האינפרא אדום!

מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©



נורה מחליפה צבעים

בשיעור זה נתעסק עם נורה שמכילה בתוכה אפשרויות לצבעים שונים (אדום, צהוב וירוק כבסיס אבל ניתן גם לערבב ביניהם וליצור צבעים שונים באותה הנורה!) את הנורה ניתן לרכוש בלחיצה [ממש פה](#)

נתחיל בחיבור הפינים השונים ל-board שלנו.
R,G,B מייצרים את הצבעים - אדום, ירוק וצהוב בהתאמה. נחבר אותם לשלושה פינים כרצונכם בתור אזור הDIGITAL (נניח ל7,6 ו8).
את הV נחבר לGND שלנו על מנת להעניק מתח כוח לרכיב שלנו...



כעת, בקוד עצמו - נגדיר את הצבעים שלנו כמשתנים גלובליים מסוג int ונעניק להם את המספרים של הפינים אליהם הם מקושרים.



מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©

נמשיך לsetup בו נגדיר את פונקציית pinModen שמקבלת שני פרמטרים. האחד שם המשתנה שלנו (אדום/צהוב/ירוק במקרה זה) וכפרמטר שני האם הוא קולט או מציג. התשובה היא מציג ולכן נכתוב כפרמטר נוסף את המילה OUTPUT.

לסיום, בתוך הloop שלנו (הפונקציה שרצה תמיד שוב ושוב לנצח...כמאין לולאה אינסופית) נוכל להגדיר פונקציית digitalWrite שמכילה שני פרמטרים. האחד את השם של הרכיב והשני האם הוא פועל או לא... (LOW או HIGH).

נסו לאתגר את עצמכם. צרו תרשים זרימה לוגי ותארו דרכו - מה התהליך המעגלי שתבצע המערכת שלכם. הקוד שעליכם יהיה לכתוב הוא: הדלקת הנורה בצבע סגול!!
פיתרון מלא לקוד בסיסי בעמוד הבא.

```
int red = 6;  
int green = 7;  
int blue = 8;
```

```
void setup(){
```

```
    pinMode(red, OUTPUT);  
    pinMode(green, OUTPUT);  
    pinMode(blue, OUTPUT);  
}
```

```
void loop(){
```

```
    digitalWrite(red, LOW);  
    digitalWrite(green, HIGH);  
    digitalWrite(blue, LOW);  
}
```



חיישן מודד מרחק

בשיעור זה, נתעסק עם סנסור 'מודד מרחק' שיכול להציג לנו פלט בצורת מספר - מהו המרחק של אובייקט כלשהו במרחב מהחיישן? (יציג בסנטימטרים) לאחר מכן, כאשר הסנסור בידיכם, תוכלו להצטרף. ראשית, על מנת לרכוש את החיישן לחצו [כאן](#) להרפתקה.

נחבר את החיישן שלנו באמצעות ארבעה חוטי ג'אמפרים:

- VCC - יתחבר לV5 באזור הPOWER בלוח שלנו
- Trigger - יתחבר לפין אקראי באזור הDIGITAL לבחירתכם (נניח ל6), המשמעות של Trigger היא קליטת התזוזה מסביב לחיישן
- Echo - יתחבר לפין אקראי באזור הDIGITAL לבחירתכם (נניח ל7), המשמעות של Echo היא שליחת גלים. על מנת להבין טוב יותר את הtrigger והecho צפו בסרטון המצורף.
- GND - יתחבר לפין הGND באזור הPOWER בלוח שלנו

כעת, בתכנות, נרצה ליצור מצב בו ברגע שנתקרב לחיישן - הדלק נורה/הפעל זמזם וכדומה.

אולם, קודם לכן, הצג את המרחק בסנטימטרים מהאובייקט הקרוב ביותר לחיישן!

מדריך ארדואינו מאתר "עולם ההייטק" המקורי ©

הקוד שלנו מתחלק לשלושה שלבים: משתנים גלובליים שדרכם נקשר את ה trigger וה echo מהלוח לתכנות.

בתוך ה setup נגדיר pinMode להתחבר אליהם וכן גם Serial.begin(9600) להפעיל את הקונסול.

בתוך ה loop נבצע שליחת גלי תדר לסביבה וקליטה ראשונית לחיישן באמצעות מספר שורות קוד מובנות. כמו גם, נקבל את המידע באמצעות פונקציית pulseIn שקולטת את הגלים. לא לשכוח להציג את המידע בתוך ה Serial וכן גם לבצע delay!

פיתרון לדוגמא בעמוד הבא.



```
int trigger = 7;
int echo = 6;

void setup(){

    Serial.begin(9600);
    pinMode(trigger, OUTPUT);
    pinMode(echo, INPUT);
}

void loop(){

    digitalWrite(trigger, LOW);
    delayMicroseconds(2);
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger, LOW);

    long duration = pulseIn(echo, HIGH);

    Serial.println(duration / 29 / 2);
    delay(1000);
}
```

חיישן מודד דציבלים

בשיעור זה, נלמד על חיישן מודד הדציבלים שתוכלו לרכוש [מכאן](#)

את החיישן הזה נוכל לחבר לפינים בלוח שלנו ולקבל פלט של מספרים עולים. המספרים שנקבל הם דציבלים, כלומר - אם החיישן יישמע קול כלשהו הוא יציג מספר לפי מד הקול ששמע. ככל שחזק יותר (צעקה) יציג מספר בSerial גבוה יותר. אנחנו נרצה לגרום אפילו למחיתא כף להדליק את הנורה שלנו.

ראשית, נחבר את הנורה האדומה שלנו (ראו ערך: שיעור Blinking Led או נורה מחליפה צבעים) ונכתוב בקוד שלנו שהיא מתחילה במצב כבוי (HIGH).

לאחר מכן, נחבר את הפינים של הסנסור שלנו למקומות הרלוונטים: GND לאזור GND לאחר מכן, כמו גם, את הV5 לאזור הV5. לסיום, את הג'אמפר של A נחבר לA0 (באזור Analog). הסיבה שהפעם אנחנו באזור האנלוג היא שאנחנו לא רוצים לקבל רק 2 ערכים (LOW וHIGH) כמו שהיה עד כה - זה מה שנותן לנו הDIGITAL. רק 2 מספרים.), אנחנו רוצים לקבל המון מספרים בסדר עולה (0 ואילך - ולכן נשתמש בANALOG - שנותן לנו מספרים רבים ולא רק שני מספרים). למי שלא סגור על ההבדלים בין דיגיטל לאנאלוג - הנורה צריכה רק 2

פרמטרים לכן נחבר אותה לדיגיטל... החיישן מודד דציבלים צריך המון פרמטרים לכן נחבר אותו לאנאלוג.

לפניכם פיתרון מלא לדוגמא:

```
int red = 6;

void setup(){
  pinMode(red, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(red, HIGH);
  int sound = analogRead(A0);
  Serial.println(sound);
  if (sound >= 50){
    digitalWrite(red, LOW);
    delay(150);
  }
}
```

