

מגמה הנדסה יישומית

מודולה 1 כיתה י'

חוברת למורה בנושא: תכנות מיקרובקר ארדואינו בסביבת Scratch

30 שעות

מחבר:

מרק טסליצקי

עורכת:

ד"ר אמונה אבו-יונס עלי

ייעוץ מדעי ופדגוגי:

ד"ר משה גרינהולץ

מפמ"ר המגמה:

שלומי אחנין

סמל מגמה: 1150

עדכון: אוגוסט תשפ"ה

כל הזכויות שמורות למשרד החינוך

מרכז המורים הארצי למקצועות הטכנולוגיים, מור-טק

הפרויקט מבוצע על ידי מוסד הטכניון עפ"י מכרז 22/11.2020

הפרויקט מבוצע עבור המזכירות הפדגוגית, משרד החינוך

החוברת יצאה לאור במימון האגף למדעים במזכירות הפדגוגית ומינהלת מל"מ המרכז הישראלי לחינוך מדעי טכנולוגי.

אין לשכפל, להעתיק, לצלם, להקליט, לתרגם, לאחסן במאגר מידע, לשדר או לקלוט בכל דרך או אמצעי אלקטרוני, אופטי או מכני או אחר כל חלק שהוא מהחומר שבחוברת זו. שימוש מסחרי מכל סוג שהוא בחומר הכלול בחוברת זו אסור בהחלט אלא ברשות מפורשת בכתב מהמוציא לאור.

תוכן עניינים

1.....	קורס Arduino ו-Tinkercard דרך Scratch
3.....	פרק א. היכרות עם Arduino - רקע
5.....	פרק ב. עבודה עם לוח חיבורים (Breadboard) ונורות LED
7.....	פרק ג. הרשמה וכניסה לאתר Tinkercad
9.....	פרק ג. שליטה בנורות LED באמצעות Scratch
13.....	פרק ד. עבודה עם כפתורים ומתגים
16.....	פרק ה. משתנים ואופרטורים לוגיים ב-Scratch ו-Arduino
19.....	פרק ו'. שימוש באותות אנלוגיים
23.....	פרק ז. שימוש בזמזם (פייזואלקטרי) ליצירת צלילים
24.....	פרק ח'. Pulse Width Modulation (PWM) ומנועי סרוו (תנועה מכנית)

נושאי הלימוד ומושגים עיקריים

1. הכרת Arduino מבחינת חומרה

- מבנה לוח החיבורים (Breadboard) והבנת קווי המתח והאדמה
- רכיבי בסיס: נגדים, לדים, חיישנים, מפעילים
- חיבור לוח Arduino לסימולטור והגדרות ראשוניות
- קריאת סכמות וחיבורי רכיבים בסביבה וירטואלית

2. הכרת Scratch

- כניסה לממשק העבודה והגדרות פרויקט חדש
- זיהוי קטגוריות הפקודות ב- Scratch המותאם ל-Arduino
- עבודה עם בלוקים לפלטים דיגיטליים ואנלוגיים
- שילוב פקודות בקרה, לולאות, ותנאים

3. הפעלת רכיבים ספרתיים (לדים, לחצנים)

- חיבור לדים עם נגדי הגנה והפעלתם בפלט דיגיטלי
- שימוש בכפתורי לחיצה (Pushbutton) ומתגים (Slide Switch)
- קריאת ערכי קלט דיגיטליים והגבת הקוד בהתאם
- תרגול בסיסי: נורת מהבהבת, שליטה בלד באמצעות לחצן

4. חיבור חיישנים ומפעילים (אקטואטורים)

- חיבור חיישני אור (LDR) ופוטנציומטרים לקריאות אנלוגיות
- הפעלת זמזמים (Passive / Active) ליצירת צלילים
- שליטה במנועי סרוו באמצעות אותות PWM
- שילוב חיישנים ומפעילים בפרויקטים אינטראקטיביים

פרק א. היכרות עם Arduino - רקע

אם פעם מחשבים היו תופסים חדר שלם, היום הם יכולים להיכנס לכיס. לא מעט מהיכולות האלה מתאפשרות בזכות עולם מדהים בשם – **Arduino** פלטפורמה פתוחה, פשוטה וחכמה, שמאפשרת לתכנת ולשלוט ברכיבים אלקטרוניים בעולם האמיתי.

Arduino - היא פלטפורמה אלקטרונית בקוד פתוח (Open Source), המבוססת על לוח בקרה קטן – לרוב בגודל של כרטיס אשראי – שבו מותקן מיקרו בקר. מיקרו בקר הוא למעשה, "המוח" של הלוח: הוא מקבל קלט, מקבל החלטות, ומפעיל פקודות בהתאם.

את לוח ה- Arduino מתכנתים באמצעות המחשב, בשפת תכנות פשוטה (דומה לשפת C) הוא מסוגל להתחבר ולהפעיל רכיבים שונים כגון נורות, חיישנים מורכבים, מנועים, פרוטוקולי תקשורת ועוד.

במילים אחרות – הוא "המוח" שאפשר להלביש על כל מיני פרויקטים, ודרכו לגרום לדברים לזוז, להאיר, לצפצף, למדוד טמפרטורה, להגיב למגע – כל מה שעולה בדמיון.

הפרויקט נולד בשנת 2005 על ידי קבוצת מהנדסים ואנשי חינוך באיטליה, במכון IVREA לעיצוב אינטראקטיבי. השם "Arduino" מגיע משם של בר-יין מקומי שבו נהגו לשבת ולחלום על רעיונות.

המטרה שלהם הייתה ברורה: **לאפשר גם לאנשים שאין להם רקע מקצועי בהנדסה – כמו מעצבים, תלמידים ומורים – לבנות בעצמם התקנים אינטראקטיביים.**

הם רצו שהמערכת תהיה:

- זולה
- פשוטה להפעלה
- נגישה לכל אחד
- פתוחה ומבוססת קוד חופשי

מאז ועד היום נמכרו **עשרות מיליוני לוחות Arduino** והם משמשים בתי ספר, מעבדות מחקר, חברות הזנק, ותלמידים בבתי ספר תיכוניים.

לוח Arduino כולל מספר רכיבים עיקריים:

1. **מיקרו-בקר לרוב מסוג ATmega328**
המוח של הלוח – שם מאוחסן הקוד, ושם מתקבלות ההחלטות.
2. **כניסות/יציאות דיגיטליות (Digital I/O)**
דרכן ניתן לשלוח ולקבל אותות בינאריים – לדוגמה, להדליק נורה או לבדוק אם כפתור נלחץ.
3. **כניסות אנלוגיות (Analog Input)**
משמשות לקבלת ערכים רציפים, כמו מדידת אור, טמפרטורה או עוצמת קול.
4. **כניסת חשמל (Power Input)**
ניתן להפעיל את הלוח דרך כבל USB מהמחשב או מסוללה חיצונית.
5. **יציאת USB**
לחיבור למחשב לצורך תכנות.

6. נורית לד קטנה: (LED)

נורית פנימית שמחוברת לרגל מספר 13 – משמשת לבדיקה מהירה של תוכניות.

שימושים של מיקרו בקר ארדואינו

1. חינוך טכנולוגי

- מתאים מאוד ללימוד מושגים כמו זרם, מתח, חיישנים, בקרה, ותכנות.
- מאפשר לתלמידים "לראות בעיניים" איך קוד מפעיל מכונה.
- מחבר בין העולם התאורטי של המדעים לבין היישום המעשי.

2. בניית פרויקטים אינטראקטיביים

- חיישני אור שגורמים למנורה להידלק כשמחשיך.
- מנועים שפותחים דלת אוטומטית כשהחיישן מזהה תנועה.
- מד חום שמציג את הטמפרטורה על מסך LCD.

3. רובוטיקה

- בניית רובוט נייד שמגיב לסביבה.
- רובוט שיכול לעקוב אחרי קו, להתחמק ממכשולים, או להגיב למחאות כפיים.

4. אוטומציה ביתית (Home Automation)

- שליטה באורות, תריסים ומזגנים באמצעות אפליקציה.
- חיישנים שמזהים נזילות מים או כניסה לא מורשית לבית.

5. אמנות דיגיטלית ואינטראקטיבית

- פסלים שמשנים צורה לפי רעש בסביבה.
- עבודות עיצוב אינטראקטיביות שמגיבות לקהל.

למה דווקא Arduino ולא כל מיקרו-בקר אחר?

- **פשטות ונוחות:** אין צורך במעגלים מסובכים, כל מה שצריך זה כבל USB ותוכנה.
- **קהילה ענקית:** אלפי פרויקטים, מדריכים, סרטונים ודוגמאות זמינות אונליין.
- **תיעוד מעולה:** המדריכים הרשמיים ברורים ומתאימים גם למתחילים.
- **מודולרי:** קל מאוד להרחיב את הלוח עם חיישנים, רכיבים נוספים (מה שנקרא "שילדים").
- **מחיר נגיש:** אפשר לקנות לוח בארדואינו מקורי או תואם ב-30–100 ש"ח בלבד.

יתרונות:

2. בדיקת מעגלים לפני בנייה קבועה

רוצים לבדוק אם המעגל שלכם עובד לפני שמרכיבים אותו סופית Breadboard? הוא המקום להתחיל בו.

3. לימוד אלקטרוניקה בצורה חווייתית

תלמידים יכולים ללמוד איך זרם זורם, איך חיבורים עובדים, ואיך מעגלים מגיבים בפועל – ולא רק בתיאוריה.

4. ניסויים מהירים וגמישים

בין אם אתם עובדים עם לדים, נגדים, חיישנים, או אפילו לוחות – Arduino הלוח מאפשר להרכיב ניסוי חדש תוך שניות.

טיפים לעבודה עם Breadboard

- השתמשו בחוטים קצרים וצבעוניים כדי להבדיל בין חיבורים שונים.
- חברו קודם את ספק הכוח ואז את הרכיבים – כדי למנוע קצר.
- בדקו את הכיוונית של רכיבים כמו דיודות, לדים או טרנזיסטורים – יש להם כיוון עבודה מסוים.

שילוב עם Arduino

Breadboard משתלב בצורה מושלמת עם לוחות ארדואינו. אפשר לחבר את יציאות הלוח ישירות ללחם החיבורים באמצעות חוטים (Jumpers) ולשלוט בלדים, מנועים, חיישנים ועוד – ללא הלחמה וללא מעגלים קבועים.

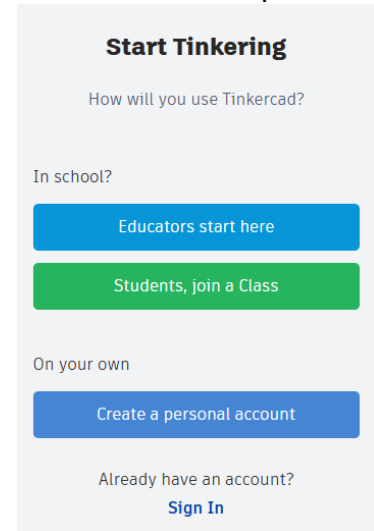
פרק ג. הרשמה וכניסה לאתר Tinkercad

היכנסו לאתר www.tinkercad.com

לחצו על "Sign Up" בצד ימין עליון של המסך:



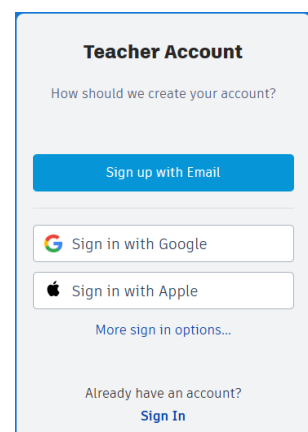
עברו למסך הבא:



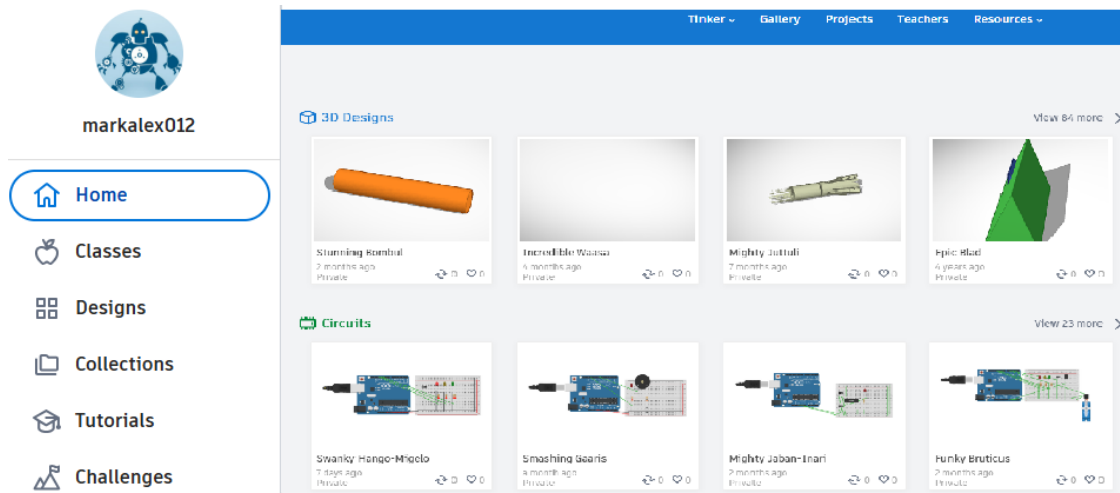
אם תרצו לעבוד עם כיתה, בחרו במצב Educators, אך שימו לב שסטודנטים יעבדו במצב שנקרא Safe Mode (ניתן לקרוא על מצב זה בקישור הבא: <https://www.tinkercad.com/help/classrooms/what-is-safe-mode>).

חותרים על ההסכם וממשיכים.

בוחרים אחד מהאפשרויות הרשמה הבאות:



היכנסו למסך ראשי:



Home מעבר למסך הראשי, בו רואים את כל העיצובים והפרויקטים שלך.

Classes ניהול כיתות וקורסים מקוונים למורים ולתלמידים.

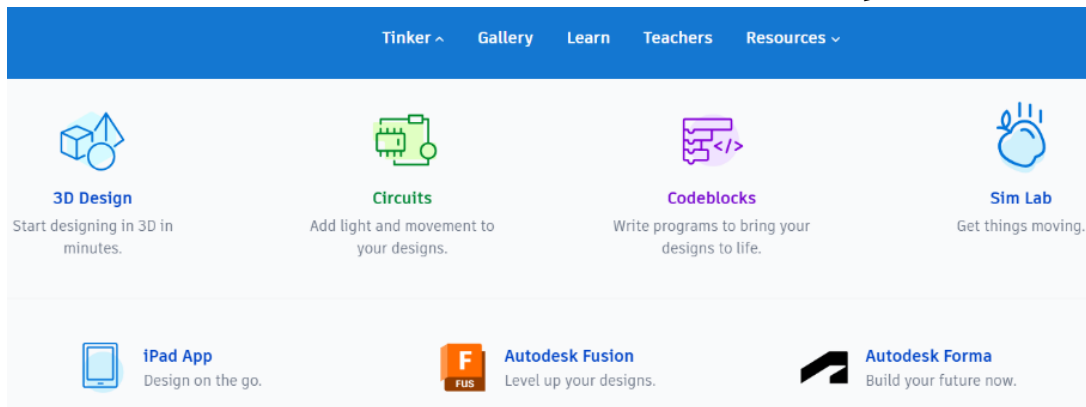
Designs הצגת כל העיצובים שיצרת, כולל תלת-ממד ומעגלים אלקטרוניים.

Collections ארגון העיצובים שלך לאוספים לפי נושאים.

Tutorials גישה למדריכים והסברים צעד אחר צעד לשימוש ב-Tinkercad.

Challenges אתגרי עיצוב ופרויקטים יצירתיים בהם ניתן להשתתף.

פקדים בשורת התפריט העליונה



Tinker מעבר למסכי עבודה ראשיים כמו 3D Design, Circuits, Codeblocks ו-Sim Lab.

Gallery צפייה ושיתוף של פרויקטים שנעשו על ידי משתמשים אחרים בקהילת Tinkercad.

Learn גישה למדריכים, שיעורים וקורסים מקוונים לשימוש בכלי Tinkercad.

Teachers משאבים וכלים ייעודיים למורים לניהול כיתות ופעילויות.

Resources קישורים ותכנים משלימים, כולל הורדות, עזרה ותמיכה.

פקדים במסך הראשי (אייקונים גדולים)

3D Design יצירת מודלים בתלת-ממד בקלות ובמהירות.

Circuits בניית מעגלים אלקטרוניים עם רכיבים כמו לדים, חיישנים ומנועים.

Codeblocks תכנות באמצעות בלוקים להפעלת אובייקטים ויצירת אנימציות.

Sim Lab ביצוע סימולציות והדמיות תנועה ואינטראקציות פיזיקליות.

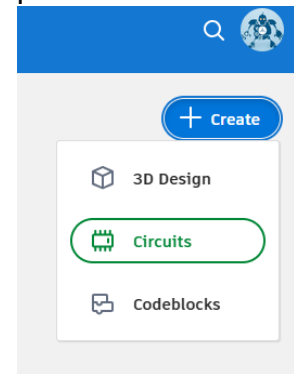
iPad App עבודה עם Tinkercad בכל מקום דרך אפליקציה ייעודית לאייפד.

Autodesk Fusion כלי מקצועי לעיצוב והנדסה מדויקת של מוצרים.

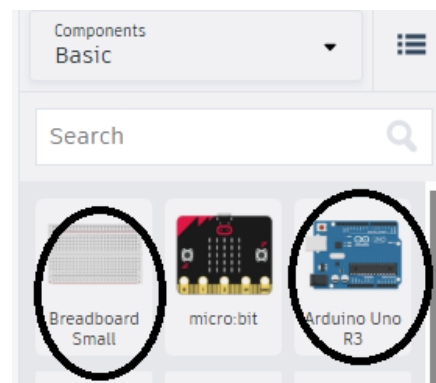
Autodesk Forma כלי לתכנון סביבתי ואדריכלות חכמה.

פרק ג. שליטה בנורות LED באמצעות Scratch

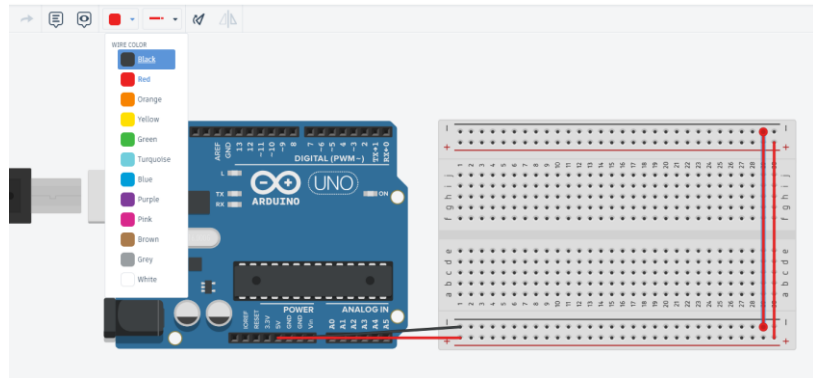
בפעם ראשונה צרו חשבון חדש עם חשבון גוגל הקיים. היכנסו לחשבון בצורה הבאה:



מתוך תפריט Basic בחרו לוח Arduino Uno ולוח חיבורים Small Breadboard.



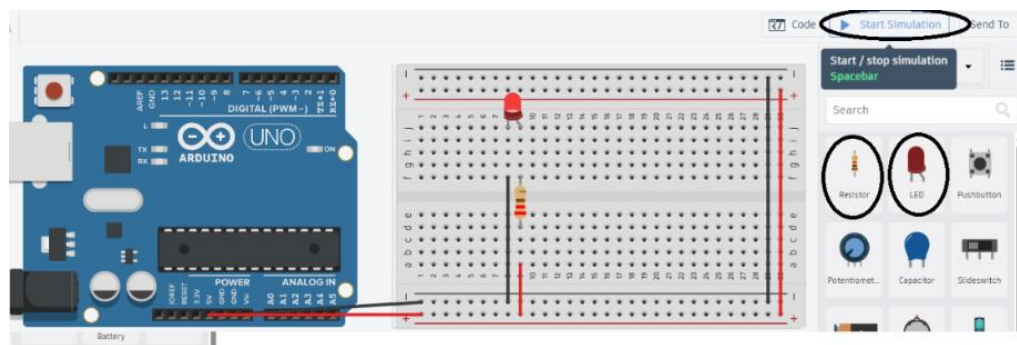
חברו הספקת מתח ואדמה משותפת. שנו צבעים של חוטים בהתאם לתפקיד (מתח – צבע אדום, אדמה צבע שחור).



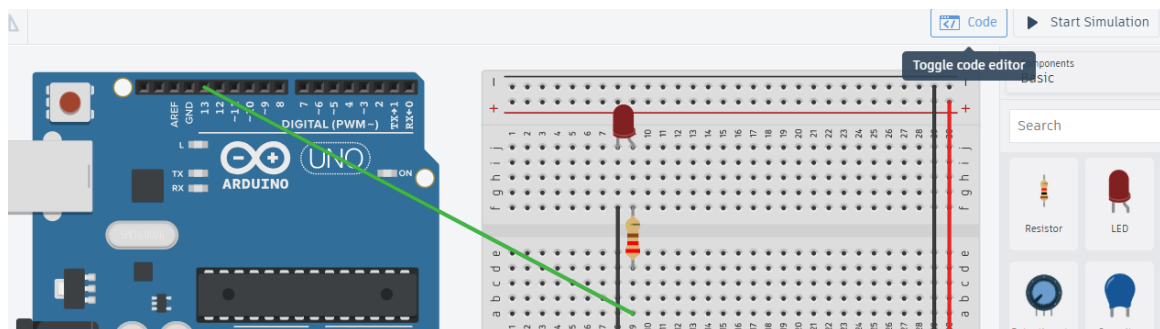
שימוש בפלטים דיגיטליים

חברו לד והפעילו הדלקה וכיבוי לד.

כדי להגביל את כמות הזרם שזורם דרך הדיודה, חברו נגד 220 אוהם. בהתחלה הפעילו לד ללא מיקרו בקר, רק כדי לבדוק עבודה תקינה של המעגל.

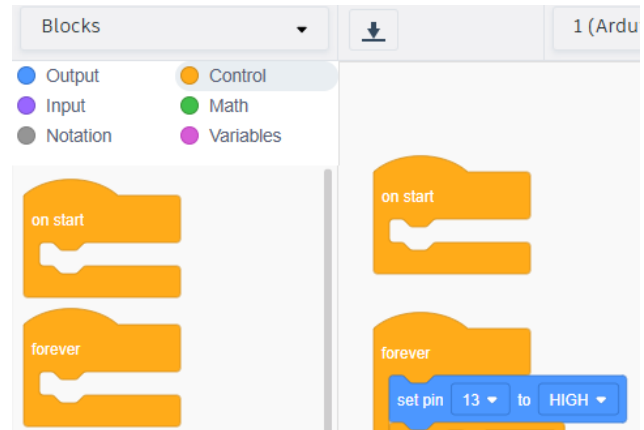


העבירו רגל הפעלת לד לאחד הפורטים, למשל לפורט 13 והפעילו קוד בשפת Scratch.



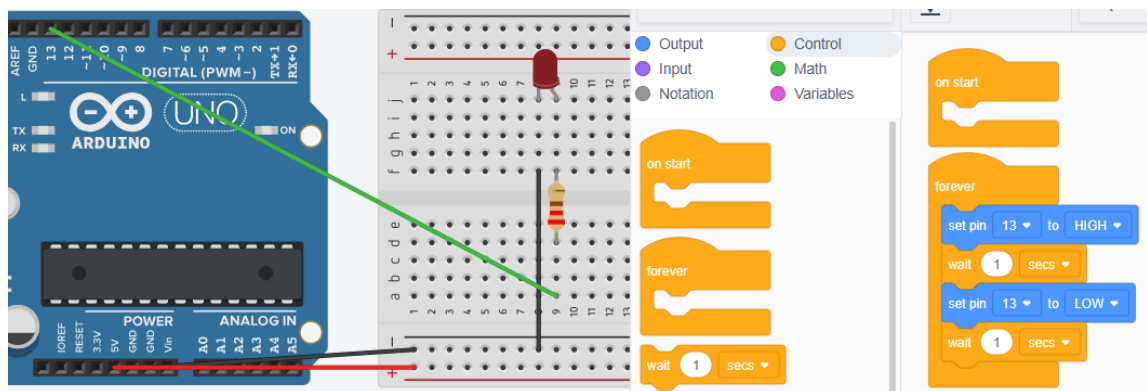
עברו לקוד בעזרת כפתור Code.

מתפריט Control במסך הראשי נבחר פונקציה on start ופונקציה (לולאה) forever.



פונקציה on start מכילה הגדרות והפעלות חד פעמיות ולולאה forever שפועלת בצורה אינסופית.

בינתיים נשאר פונקציה on start ריקה ונוסיף הדלקת וכיבוי לד מתפריט Output (צבע כחול) והשהייה wait מתפריט Control.



תרגיל 1

בנו רמזור בצורה הבאה. חברו 3 לדים: אדום, צהוב, ירוק והפעילו אותם בצורה הבאה:

הפעילו צבע אדום במשך 3 שניות.

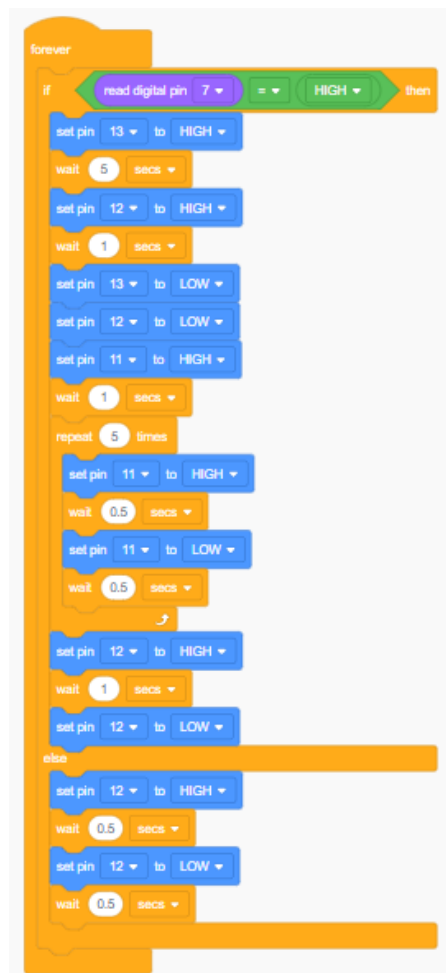
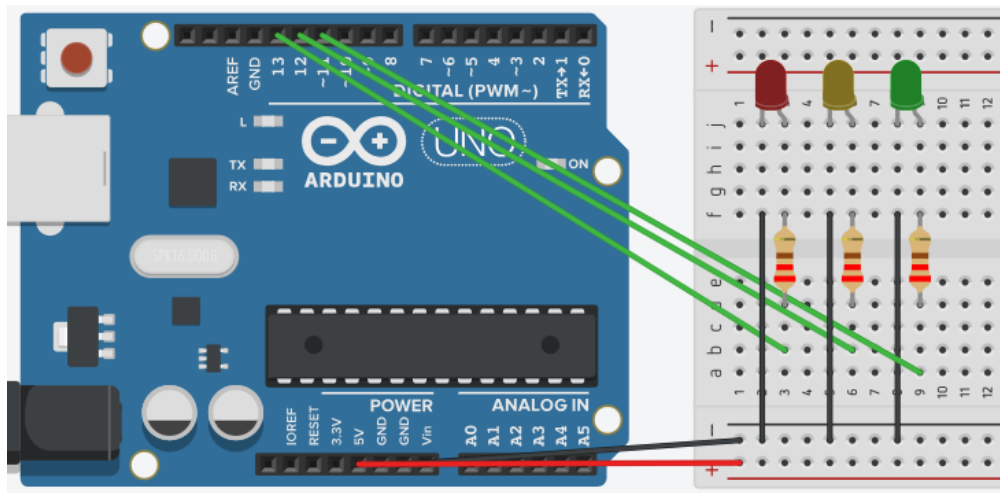
הפעילו לד צהוב יחד עם צבע אדום למשך שנייה.

כבו צבע אדום וצבע צהוב והפעילו לד ירוק למשך 3 שניות.

הדליקו וכבו צבע ירוק 5 פעמים עם השהייה של חצי שנייה.

הדליקו צבע צהוב למשך שנייה ואז כבו אותו. הכול מתחיל מהתחלה.

פתרון תרגיל 1



פרק ד. עבודה עם כפתורים ומתגים

מטרת הפרק

מטרת הפרק היא ללמד את התלמידים כיצד לחבר ולהשתמש בכפתורי לחיצה (Pushbutton) ובמתגים (Slide Switch) במעגלים אלקטרוניים עם Arduino וכיצד לשלוט בפעולות באמצעות סביבת Scratch. הפרק מדגיש את ההבנה המעשית של עקרון הקלט הדיגיטלי, תרגול חיבורים נכונים, וכתובת קוד בסיסי שמגיב לשינוי במצב הכפתור.

הנחיות למורה

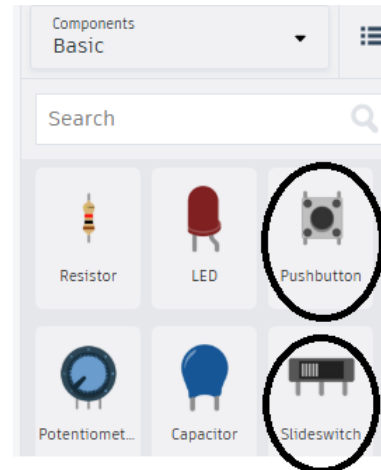
- התחילו בהסבר קצר על סוגי הכפתורים וההבדל ביניהם (מצב יציב מול מצב רגעי).
- הדגו חיבור פשוט של כפתור ללוח Breadboard ולארדואינו כולל חיבור נגד במידת הצורך.
- הראו כיצד קריאת הקלט הדיגיטלי מופיעה ב-Scratch והסבירו את משמעות הערכים "0" ו-"1".
- בצעו יחד עם התלמידים דוגמה של שליטה בלד לפי מצב הכפתור.
- הציעו לתלמידים אתגרים להמשך, כגון שילוב כפתור עם זמזם או עם מנוע סרוו.

מה יכול להופיע בשיעור

- דיאגרמות חיבור ברורות עם סימון הצבעים של החוטים.
- קטעי קוד Scratch מוכנים ודוגמאות לשינויים אפשריים.
- תרגיל חקר קטן: מה קורה אם משנים את התנאי בקוד, או מחליפים את הפלט (לד ← זמזם).
- משימות להרחבה: שילוב שני כפתורים לבקרה כפולה, או הפעלה רק בסדר פעולות מסוים.

איך עובד כפתור?

ישנם 2 סוגי כפתורים ב-Tinkercad:

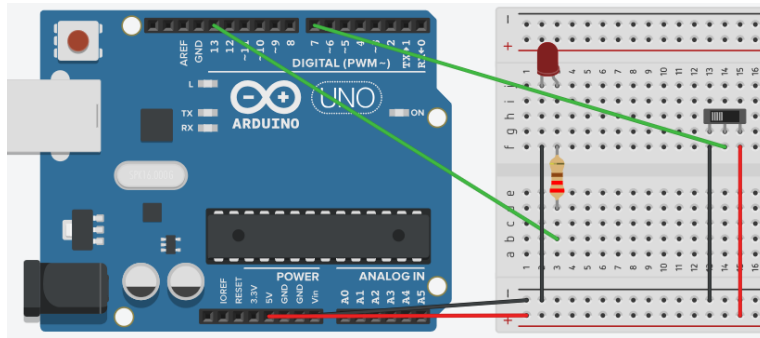


ההבדל ביניהם:

Slide Switch – נשאר באותו מקום איפה שאנו משאירים אותו. Pushbutton נשאר במצב הלא יציב כל עוד אנו מחזיקים אותו – ברגע משחררים אותו כפתור חוזר למצב יציב.

שליטה בפעולות דרך כפתור ב-Scratch

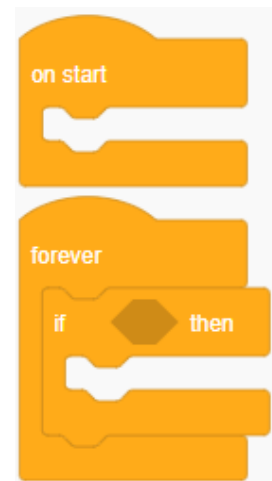
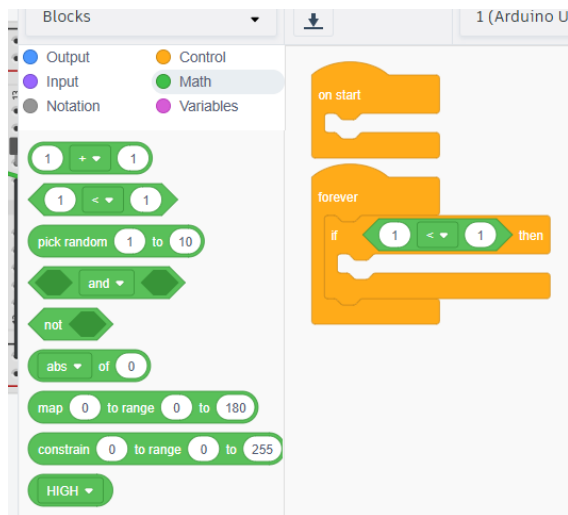
חברו קודם Slideswitch בצורה הבאה:



כתבו קוד חדש:

הוסיפו תנאי מתוך תפריט Math:

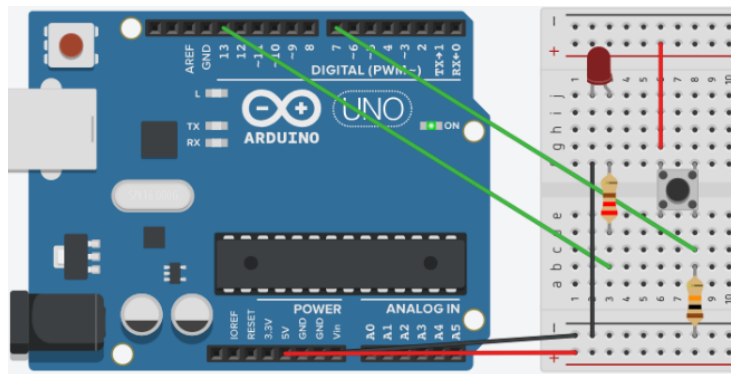
הכניסו תנאי if מתוך תפריט Control ללולאה:



עברו לתפריט Input ובחרו read digital pin והוסיפו קוד של הדלקת וכיבוי של לד. שימו לב על שינוי תנאי – סימן "=".



הפעילו את הקוד, אם כפתור נמצא במצב "0" – לד יהיה כבוי, אם תעבירו כפתור למצב "1" – לד מהבהב. שימוש בכפתור לחיצה.



כל עוד לא לוחצים על הכפתור – לד יהיה כבוי, אם לוחצים ומחזיקים את הכפתור – לד מהבהב. הקוד נשאר אותו קוד כמו בדוגמה עם Slideswitch.

פרק ה. משתנים ואופרטורים לוגיים ב-Arduino ו-Scratch

מטרת הפרק

מטרת פרק זה היא להקנות לתלמידים הבנה מעשית של מושגי משתנים ואופרטורים לוגיים בסביבת Scratch בשילוב עם Arduino וללמד כיצד ניתן להשתמש בהם כדי לשמור מידע, לבצע חישובים, ולשלט בפעולות המבוססות על תנאים.

הנחיות למורה

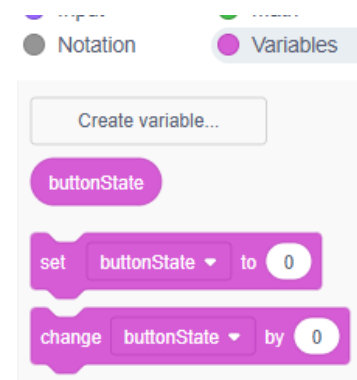
- התחילו בהסבר מהו משתנה, לשם מה משתמשים בו, ואיך מגדירים אותו ב-Scratch.
- הדגישו יצירת משתנה חדש והציגו ערכו בזמן ריצה (לדוגמה: ספירת לחיצות על כפתור).
- הציגו סוגי אופרטורים לוגיים ($>$, $<$, $=$) ושילבו בתנאים (if) לשליטה בפעולות.
- המחישו שימוש באופרטורים לוגיים מורכבים (and, or, not) לפרויקטים מתקדמים יותר.
- בצעו הדמה של תרגיל פשוט שבו משתנה משמש לקבלת החלטות בתכנית, למשל הפעלה של רכיב רק לאחר מספר לחיצות מסוים.

מה יכול להופיע בשיעור

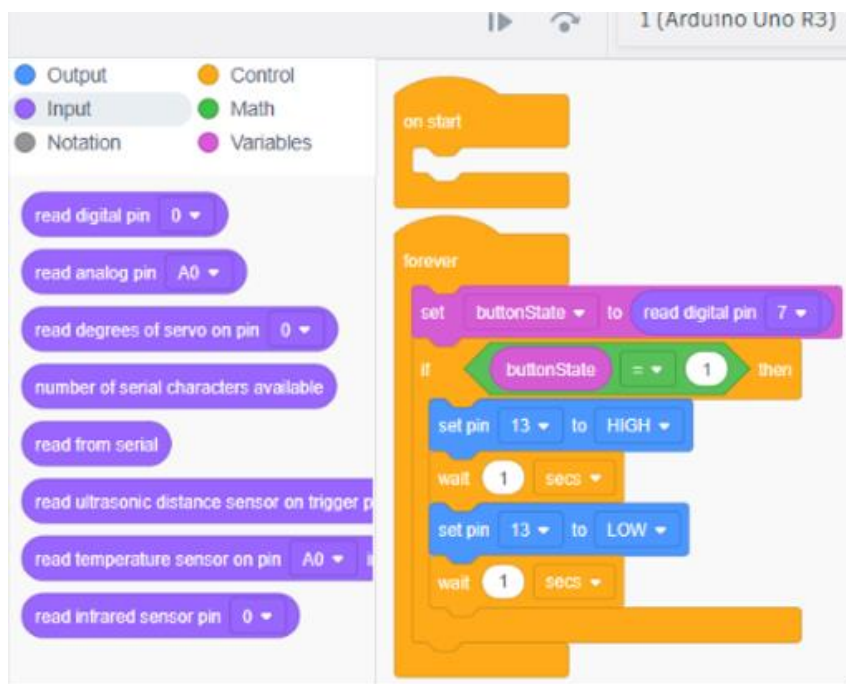
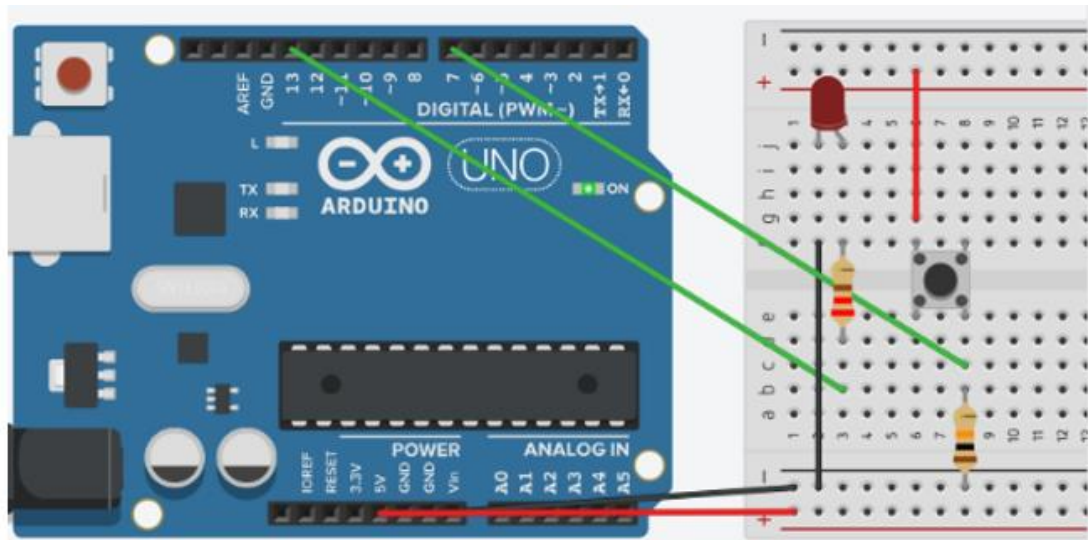
- דוגמאות קוד עם משתנים ב-Scratch המשלבים רכיבי ארדואינו (לד, זמזם, חיישן).
- טבלה קצרה עם פירוט סוגי האופרטורים הלוגיים והשימושים שלהם.
- משימת תרגול: יצירת "מונה לחיצות" שמדליק לד רק כאשר המונה הגיע לערך מסוים.
- הרחבה: שילוב חיישן אור עם משתנה – הפעלת פעולה כאשר הערך הממוצע של המדידות נמוך/גבוה מסף מוגדר.
- אתגר לתלמידים מתקדמים: שילוב שני משתנים ותנאים מורכבים (לדוגמה: חיישן אור + מספר לחיצות כפתור) להפעלה של אקטואטור.

שמירת מידע באמצעות משתנים

הגדירו משתנים כדי לשמור מצבים ומספרים בתפריט Variables.

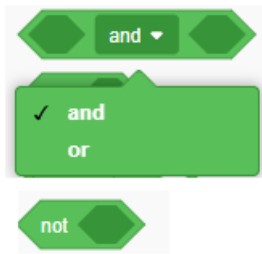
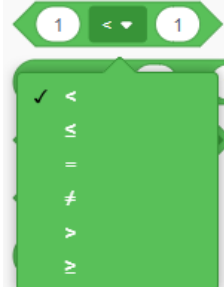
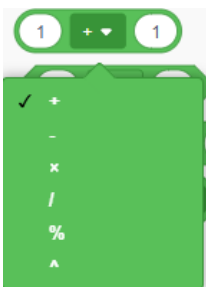


צרו משתנה חדש בשם buttonState וכתבו קוד חדש למעגל שעשיתם בעבר:



שימוש בתנאים לשליטה בפעולות

בדקו את השילוב תנאים בסיסיים ביותר לשילוב תנאים.

שילוב תנאים:	קבלת מספר אקראי:	השוואות:	פעולות מתמטיות:
			

כל פעם שתלחצו על הכפתור תגדילו ערך של משתנה ב-1 ע"י פקודה:



פרק ו'. שימוש באותות אנלוגיים

אחד המרכיבים החשובים ביותר במיקרו-בקר Arduino הוא ה- **ADC – Analog to Digital Converter** או בעברית: ממיר אנלוגי לדיגיטלי. זהו רכיב שמאפשר ללוח לקרוא מתח משתנה (כמו טמפרטורה, אור, לחץ) ולהפוך אותו לערך מספרי שהמחשב יכול להבין ולעבד.

מה זה אות אנלוגי, חישוב רזולוציה והמרת מתח לערך דיגיטלי
אות אנלוגי הוא אות רציף – לדוגמה:

- מתח שיכול להיות 1.27V, 3.59V וכו'.
- חיישן טמפרטורה שנותן מתח שמשתנה לפי חום.
- פוטנציומטר שמשנה את המתח בהתאם לסיבוב.

אבל מחשב וגם Arduino מבינים רק מספרים בינאריים – ולכן יש צורך להמיר את המתח האנלוגי לערך דיגיטלי ע"י הפעולה **ADC**.

איך ה-ADC עובד ב-Arduino?

בלוחות כמו **Arduino UNO** יש **ADC ברזולוציה של 10 ביט**. זה אומר שעבור 10 ביט ישנם 2^{10} או 1024 אפשרויות.

- ממיר ADC מחלק את המתח הכניסה במספר רמות (1024).
- כל רמה מיוצגת במספר בין 0 ל-1023.

כך:

- ערך $0 \rightarrow 0V$
- ערך $512 \rightarrow 2.5V$
- ערך $1023 \rightarrow 5V$

מה משפיע על הדיוק?

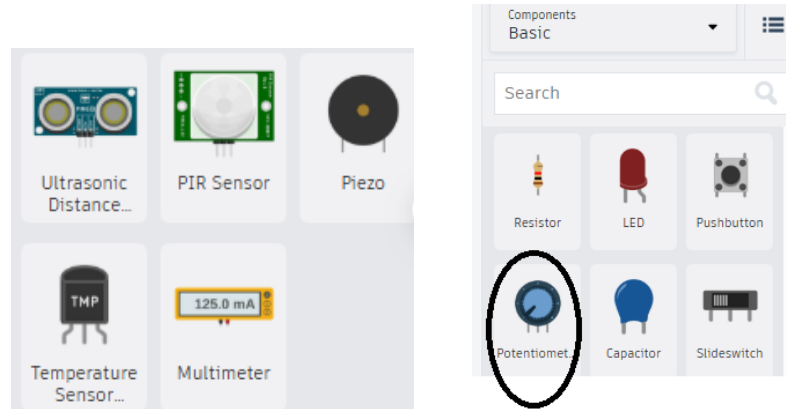
1. **מתח ייחוס לא מדויק:** ב-UNO ברירת המחדל היא 5V, אבל במציאות זה עלול להיות 4.9V או 5.1V.
2. **רעשים חשמליים:** חוטים ארוכים, מקורות מתח לא יציבים וכו'.
3. **חיישנים לא ליניאריים:** ייתכן שהחיישן לא מספק מתח מדויק יחסית לערך הנמדד.
4. **טווח קריאה:** ניתן להגדיר Vref אחר, לדוגמה 3.3V או 1.1V, כדי לקבל רזולוציה גבוהה יותר בטווחים נמוכים.

שימושים נפוצים לקריאת אות אנלוגי

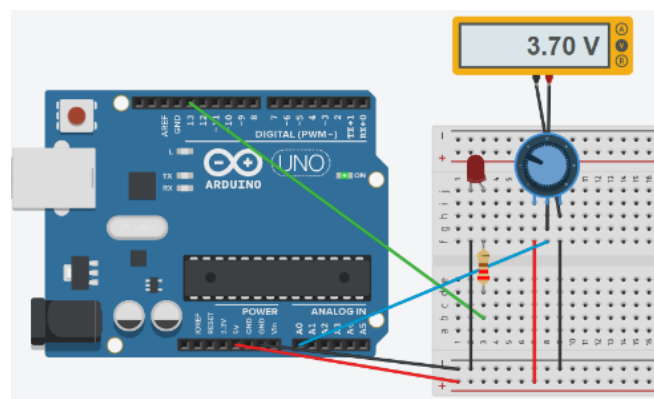
- קריאת ערכים מחיישני טמפרטורה, אור, לחץ, גז.
- מדידת ערכי כניסות ממתגים משתנים (פוטנציומטרים, חיישני מגע).
- ניטור מתח מסוללות.
- מדידות מתקדמות באמצעות מחלקי מתח.

חיבור פוטנציומטר לפורט אנלוגי

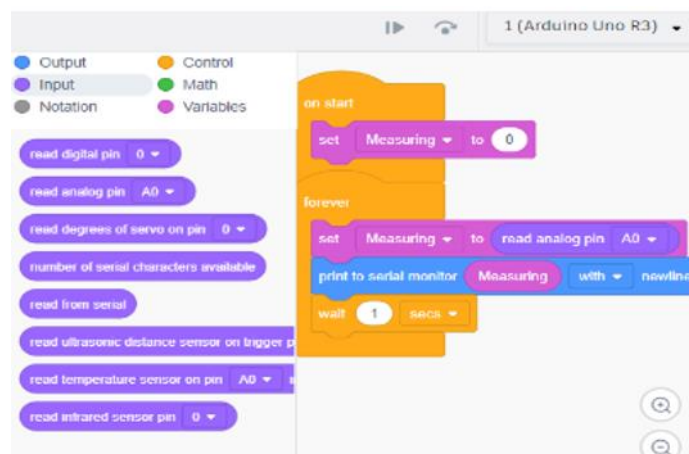
בחרו נגד משתנה מתפריט בסיסי.



חברו רגל שמאלית להספקת מתח, רגל ימנית לאדמה ורגל אמצעית לפורט אנלוגי A0. הוסיפו רב מודד למדידת מתח על פוטנציומטר מאותו תפריט.



כתבו קוד ומדדו ערך אנלוגי של פוטנציומטר:



בצעו מדידות ובנו טבלה:

מתח	ערך במסך סריאלי
3.7 V	757
5 V	1023
0 V	0
1 V	205
2.5 V	512

מדידת אור עם חיישן

היכולת של מכשיר לזהות כמה אור יש בסביבה שלו – היא כלי חזק. רוצים שווילון ייסגר כשהשמש חזקה? שנורת לילה תידלק כשהחדר חשוך? בשביל זה בדיוק נשתמש ב־חיישן אור עם **Arduino**.

מה זה חיישן אור?

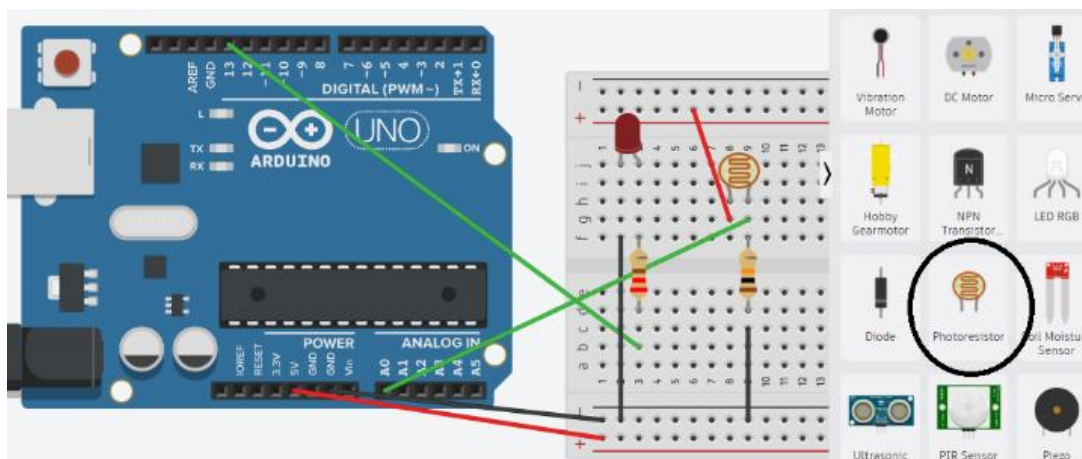
חיישן אור – (Light Sensor) ובעברית: **חיישן תאורה** – הוא רכיב שמודד את עוצמת האור בסביבתו. החיישן הנפוץ ביותר בלמידה בסיסית הוא **LDR** ראשי תיבות של **Light Dependent Resistor** כלומר: נגד שתלוי בעוצמת האור.

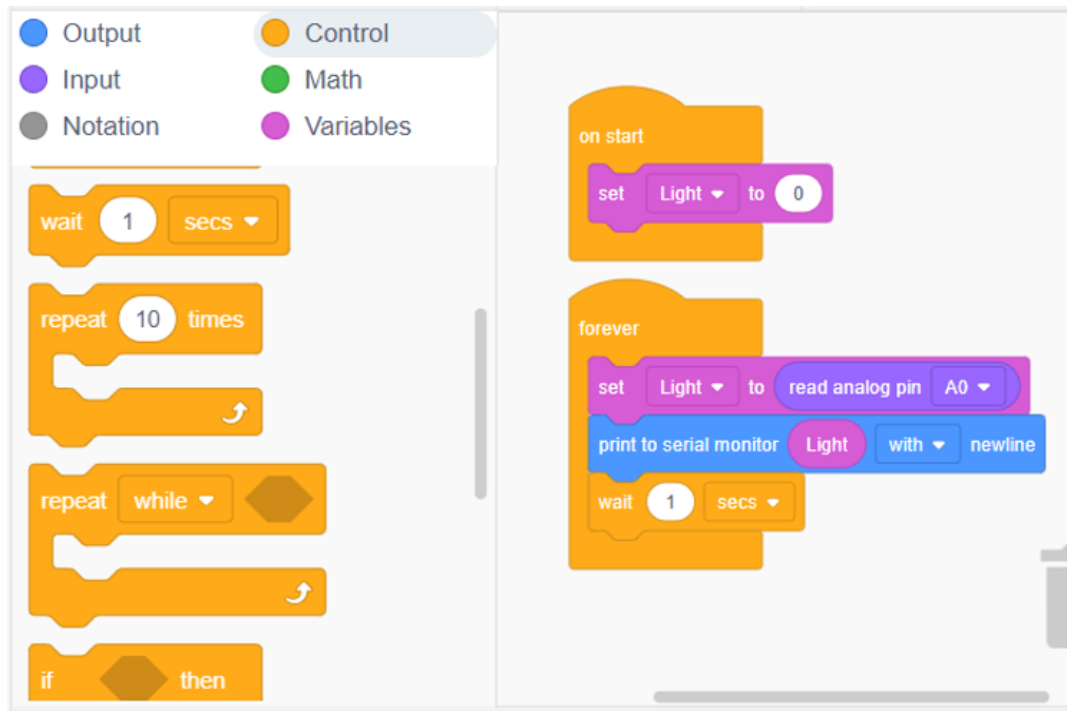
- כשיש הרבה אור, ההתנגדות של ה- **LDR** נמוכה.
- כשחשוך, ההתנגדות גבוהה.

איך מחברים את ה- **LDR** לארדואינו?

כדי לקרוא את הערך מ-**LDR** יש לבנות מחלק מתח פשוט.

- חברו את ה- **LDR** בטור עם נגד רגיל (לדוגמה 10 קילו אוהם).
- בין נקודת החיבור של שניהם – קחו **קריאת מתח אנלוגית** באמצעות פקודה `analogRead()`





פרק ז. שימוש בזמזם (פייזואלקטרי) ליצירת צלילים

הפקת צלילים עם זמזם

הזמזם (Buzzer) הוא רכיב קטן ופשוט שמסוגל להפיק צלילים – צפצופים, אזעקות, מגנינות – והכול בשליטה מלאה של קוד בארדואינו. זהו אחד הרכיבים האלקטרוניים הפופולריים ביותר בפרויקטים התחלתיים, בעיקר בזכות העובדה שהוא גם זול, גם קומפקטי וגם מרעיש.

בפרק זה נסביר מה זה זמזם, איך מחברים אותו, איך שומרים על שליטה מלאה בעוצמה ובגובה הצליל, ונסיים גם בקוד שמנגן מגנינה אמיתית.

מה זה Buzzer?

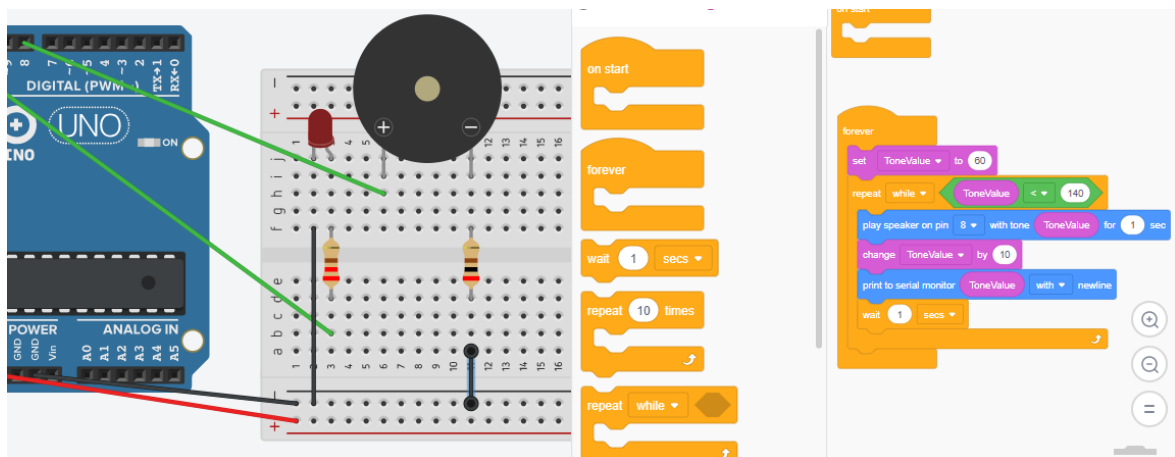
ישנם שני סוגי זמזמים עיקריים:

1. **Active Buzzer** משמיע צליל קבוע כשהוא מקבל זרם (כמו כפתור של מיקרוגל). לא ניתן לשלוט על התדר.
2. **Passive Buzzer** רכיב שמשמיע צליל בהתאם לתדר שאנחנו שולחים אליו. זה מאפשר הפקת מגנינות ואפקטים. בכדי לגנון מגנינה (ולא רק צפצוף פשוט) יש להשתמש ב-Passive Buzzer.

איך מחברים את הזמזם?

- הרגל הארוכה (האנודה, +) – מתחברת לפין דיגיטלי של Arduino, למשל D8.
 - הרגל הקצרה (הקתודה, -) – לאדמה (GND).
- ניתן להוסיף נגד של 100Ω – 220Ω להגנה.

שימו לב שבסימולטור Tinkercad צפצפה עובדת בתחום בין 40 לבין 140 הרץ, למרות שבפועל תחום השמיעה יותר רחב.



פרק ח'. Pulse Width Modulation (PWM) ומנועי סרוו (תנועה מכנית)

בעולם האלקטרוניקה, המתח שיסופק לרכיבים משפיע על איך הם פועלים. לדוגמה, לד שניתן לו יותר מתח – יאיר חזק יותר. בעת שימוש בארדואינו, מתברר שזה לא אינו פשוט "לשלוח" מתח משתנה אמיתי – לפחות לא באופן ישיר. בכדי לבצע זאת ע"י PMW אחד הכלים החזקים והחכמים בארדואינו.

מה זה בעצם PWM?

PWM היא טכניקה לשליטה בממוצע של המתח שמגיע לרכיב – באמצעות הפעלה וכיבוי מהירים מאוד של אות דיגיטלי.

- האות נשאר תמיד או ב-"0" לוגי (0 V) או ב-"1" לוגי (5 V).
 - ניתן לשלוט בכמה זמן מתוך כל מחזור האות נמצא ב-HIGH
- זה יוצר אפקט שנראה כמו "מתח ממוצע" משתנה, למרות שלמעשה מדובר רק בהפעלה וכיבוי מהירים מאוד.

מה זה Duty Cycle ?

Duty Cycle הוא המונח החשוב ביותר ב-PWM, והוא אחוז הזמן שבו האות נמצא במצב HIGH (דלוק) בכל מחזור.

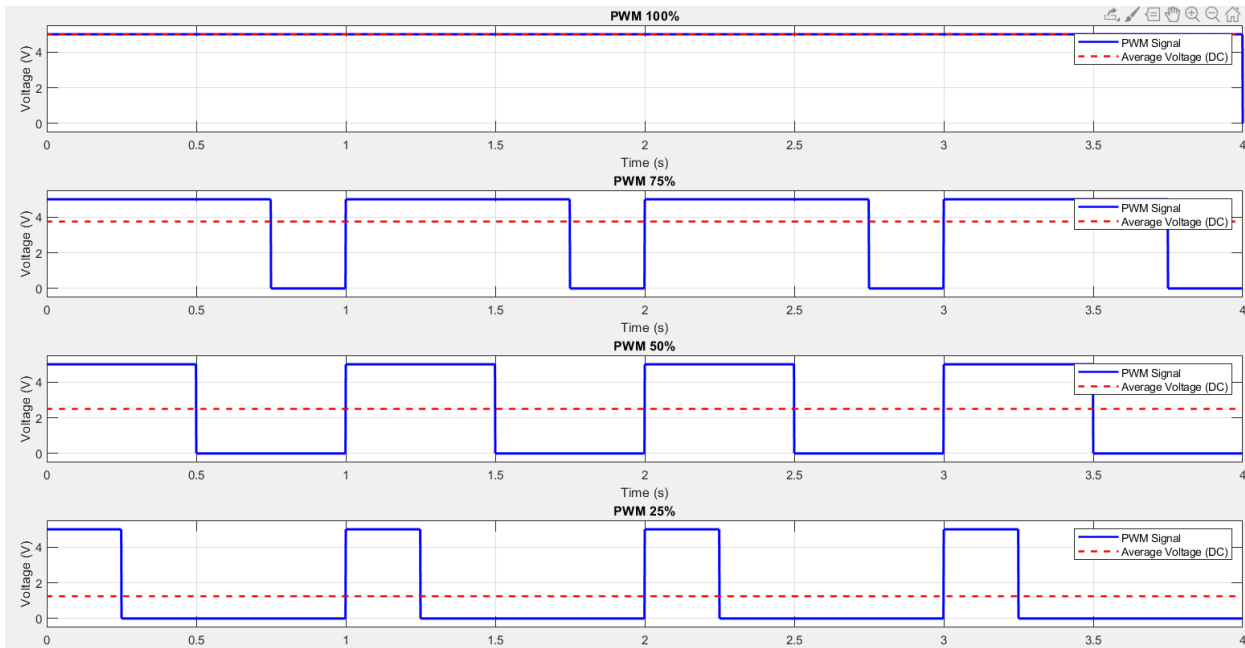
לדוגמה:

- Duty cycle של 0% – כל הזמן כבוי - ממוצע 0V
 - Duty cycle של 50% – חצי מהזמן דלוק, חצי כבוי - ממוצע 2.5V
 - Duty cycle של 100% – כל הזמן דלוק - ממוצע 5V
- ככל שה-Duty Cycle גבוה יותר, המתח הממוצע שמגיע לרכיב גבוה יותר.

PWM לעומת אות אנלוגי אמיתי

מאפיין	PWM	אות אנלוגי אמיתי
סוג אות	דיגיטלי (פולסים)	רציף
מתח	רק 0V או 5V (ON/OFF)	כל ערך בין 0V ל-5V
רזולוציה	8 ביט (0-255) ב-Arduino UNO	תלוי ב-ADC או DAC
צריכת חשמל	משתנה לפי duty cycle	רציף
שימושים	שליטה על לדים, מנועים, מהירות הגברה, סאונד, DAC מדויק	

בגרפים הבאים ניתן לראות צורות שונות של אות PWM. בגרף ראשון מקדם הוא 100% ומתח ממוצע הוא 5 V, בגרף שני 75% ומתח ממוצע הוא 3.75 V, בגרף שלישי 50% עם מתח ממוצע 2.5 V ובגרף האחרון 25% עם מתח ממוצע 1.25 V.



שימושים נפוצים ל-PWM

1. שליטה על בהירות של לד

- לד מגיב ישירות לממוצע הזרם שעובר דרכו – לכן שליטה ב-PWM מאפשרת להבהיר או לעמעם אותו.

2. שליטה על מהירות מנועים

- מנועי DC מגיבים למתח – על ידי שינוי ה־duty cycle של PWM ניתן לשלוט על מהירות הסיבוב.

3. שליטה בעוצמת צליל

- ניתן להפיק צלילים באמצעות פונקציית `tone()` אך גם `analogWrite()` יכול לשמש ליצירת צליל גס.

4. שליטה על מערכות חכמות

- אורות דינמיים, סירנות, מנועים של רובוטים, מערכות השקיה ועוד.

שימו לב שלא כל הפורטים ספרתיים מתאימים להפעלת PWM – רק האלה שלפני מספר פורט יש סימן "~":

3, 5, 6, 9, 10, 11

איך עובד מנוע סרוו?

אם אי פעם בניתם רובוט, זרוע מכנית, רכב ממונע קטן או אפילו פתחת דלת אוטומטית – כנראה שתשתמשו במנוע סרוו (Servo Motor). מנוע סרוו הוא אחד הרכיבים השימושיים והנפוצים ביותר בעבודה עם Arduino ופרויקטים אלקטרוניים בכלל. אבל מה הופך אותו לכל כך שימושי? ואיך הוא בעצם עובד? בפרק זה נבין מהו מנוע סרוו, מה מבדיל אותו ממנועים אחרים, איך שולט בו, ואיך משלבים אותו בפרויקטים חכמים.

מהו מנוע סרוו?

מנוע סרוו הוא מערכת שלמה הכוללת:

1. **מנוע DC קטן** – שמסובב את הציר.
 2. **תיבת הילוכים** – שמאטה את הסיבוב ומספקת מומנט גבוה.
 3. **מערכת חישה פנימית** (פוטנציומטר) – שמודדת את זווית הציר.
 4. **בקר אלקטרוני קטן** – שמבצע את הבקרה הפנימית.
- כל החלקים האלה משולבים בתוך קופסה קומפקטית אחת.
- בשונה ממנוע רגיל שמסתובב כל הזמן, מנוע סרוו מיועד להזיז את הציר לזווית מסוימת ולשמור עליה.

סוגים של מנועי סרוו

- **סרוו רגיל (Standard Servo)** – מוגבל לזווית של כ- 0° עד 180° .
 - **סרוו רציף (Continuous Rotation)** – מסתובב כמו מנוע רגיל, אך מקבל פקודות כמו סרוו (למשל: מהירות וכיוון).
 - **סרוו מדויק (High Torque, High Precision)** – מיועד לפרויקטים תעשייתיים וכולל חיישנים מתקדמים יותר.
- בפרויקטים חינוכיים ועם Arduino משתמשים לרוב בסרוו רגיל, לדוגמה: SG90 או MG996R.

איך שולט בסרוו?

במקום לשלוח לו מתח משתנה או זרם כמו למנוע רגיל, שולחים לו אות – **PWM (Pulse Width Modulation)** כלומר, פולסים של אות דיגיטלי שמציין את הזווית הרצויה.

איך זה נראה בפועל?

- **אות ברוחב של 1 מילישנייה** – הציר יעמוד על 0° .
 - **1.5 מילישנייה** – 90°
 - **2 מילישנייה** – 180°
- האות חוזר כל 20 מילישניות, כלומר בתדר של 50Hz.
- הבקר הפנימי בתוך הסרוו קורא את האות ומזיז את הציר לזווית המתאימה.

מבט פנימי – מה קורה בתוך הסרוו?

בתוך המנוע נמצא **פוטנציומטר**, שמחובר ישירות לציר הסיבוב. הפוטנציומטר מודד את הזווית הנוכחית ומדווח עליה לבקר הפנימי. כאשר מתקבל אות PWM חדש, הבקר משווה בין הזווית הנוכחית לבין הזווית הרצויה ומפעיל את המנוע עד שהשניים שווים.

כך מתקבלת **בקרה סגורה (Closed Loop)** – כלומר, כל הזמן יש השוואה ותיקון.

יצירת תנועה עם שליטה ב-Scratch

מנוע סרוו הוא מנוע קטן עם מערכת בקרה פנימית שמאפשרת לסובב אותו לזווית מסוימת בצורה מדויקת.

איך הוא בנוי?

- בפנים יש מנוע DC קטן.
- אליו מחוברת מערכת גלגלי שיניים שמפחיתה מהירות ומגדילה מומנט.
- יש גם פוטנציומטר (נג'ד משתנה) שמודד כל הזמן את הזווית הנוכחית.
- בקר פנימי משווה בין הזווית הרצויה (שמגיעה מהארדואינו) לבין הזווית האמיתית, ואם יש הבדל – הוא מזיז את המנוע עד שהסרוו מגיע למקום הנכון.

איך שולטים בו?

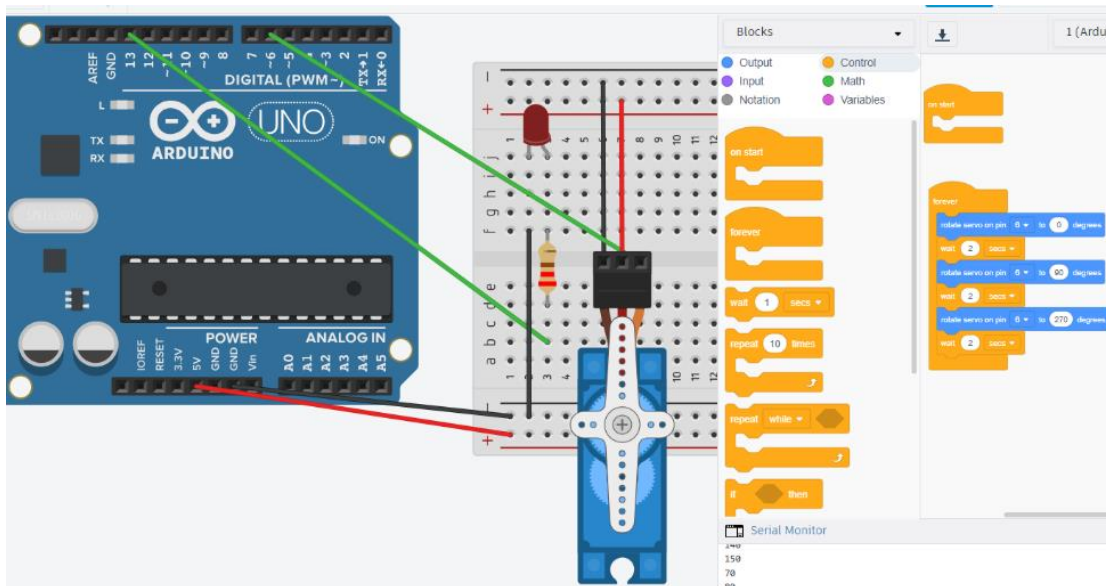
- שולחים אליו אות PWM (פולסים חשמליים מחזוריים).
- אורך הפולס למשל קובע את הזווית:

$$\sim 1\text{ms} \rightarrow 0^\circ$$

$$\sim 1.5\text{ms} \rightarrow 90^\circ$$

$$\sim 2\text{ms} \rightarrow 180^\circ$$

זה שונה ממנוע רגיל שממשיך להסתובב בלי שליטה עד שמפסיקים את המתח.



פרויקטון קטן:

הוסיפו חיישן קרבה (PIR) וכתבו קוד, כאשר משיהו מתקרב – מנוע מסתובב, אם אין אף אחד – מנוע חוזר למצב התחלתי.

