

מגמה הנדסה יישומית

מודולה 1 כיתה י'

חוברת לתלמיד בנושא: תכנות מיקרובקר ארדואינו בסביבת Scratch

30 שעות

מחבר:

מרק ססליצקי

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
2	פרק א': היכרות עם Arduino Tinkercad
6	פרק ב': עבודה עם לוח חיבורים (Breadboard) ונורות LED
8	פרק ג': שליטה בנורות LED באמצעות Scratch
15	פרק ד': עבודה עם כפתורים ומתגים
18	פרק ה': משתנים ואופרטורים לוגיים ב-Scratch + Arduino
21	פרק ו': שימוש באותות אנלוגיים
26	פרק ז': שימוש בזמזם (פיזואלקטרי) ליצירת צלילים
28	פרק ח': PWM ומנועי סרוו (תנועה מכנית)
34	סיכום

קורס Arduino ו-Tinkercard דרך Scratch

חוברת זו נכתבה במיוחד עבור תלמידי כיתה י' במגמת הנדסה יישומית, והיא מיועדת להוביל אתכם צעד-צעד לעולם מרתק של אלקטרוניקה ותכנות באמצעות ארדואינו, סימולטור Tinkercad ושפת Scratch.

במהלך הלמידה תבנו מעגלים אלקטרוניים פשוטים ותפעילו רכיבים כמו לדים, כפתורים, חיישנים ומנועים – והכול בעזרת תכנות חזותי בבלוקים. השילוב בין תיאוריה לפרקטיקה יאפשר לכם לראות במו עיניכם איך רעיון הופך למערכת שעובדת בפועל: נורה שנדלקת, זמזם שמשמיע צליל, מנוע שמסתובב או חיישן שמגיב לאור.

החוברת מחולקת לפרקים קצרים וברורים. בכל פרק תמצאו:

- **הסבר תאורטי** – מושגים בסיסיים שצריך להכיר.
 - **תרגול מעשי** – בניית מעגלים והפעלה באמצעות קוד.
 - **פרויקטים לדוגמה** – דוגמאות המדגימות יישום יצירתי של הידע.
 - **שיעורי בית ואתגרים** – משימות לחיזוק והעמקת ההבנה.
- המטרה היא לא רק ללמוד טכנולוגיה, אלא גם לפתח **חשיבה יצירתית, יכולת פתרון בעיות ועבודה עצמאית** – מיומנויות חשובות בכל תחום הנדסי וטכנולוגי.

מטרה: ללמד את התלמידים כיצד להשתמש בארדואינו לבנות מעגלים ב-Tinkercad ולשלוט ברכיבים אלקטרוניים דרך Scratch.

נושאי הלימוד: יסודות האלקטרוניקה, חיבור חיישנים ומפעילים (אקטואטורים), תכנון מעגלים, ותכנות באמצעות בלוקים.

פרק א': היכרות עם Arduino Tinkercad

המשימות המרכזיות בפרק זה:

1. היכרות והבנה

- למידת עקרונות Arduino איך הוא בנוי ולמה משתמשים בו.
- הכרת סביבת Tinkercad והבנת למה היא כל כך נוחה ללמידה ולניסוי.

2. עבודה מעשית

- כניסה ל-Tinkercad עם חשבון אישי.
- זיהוי רכיבים בסיסיים: לוח Arduino לוח חיבורים (Breadboard) לדים, נגדים וחיישנים.
- ביצוע חיבור פשוט של לוח Arduino בסביבה הווירטואלית.

3. תרגול והתנסות

- הרצת דוגמה בסיסית (לד שמדליק ונכבה).
- התנסות בבניית מעגל ראשון ובדיקה שהוא פועל כראוי בסימולטור.

4. איך להתקדם נכון

- אם משהו לא ברור – לשאול, ולא להשאיר סימני שאלה פתוחים.
- לשמור את הפרויקטים שלך ב-Tinkercad כדי לחזור אליהם בהמשך.
- התכונות לפרקים הבאים – העמקת שם בתכנות, חיישנים ואקטואטורים.

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.

בין אם הינכם תלמידים בכיתה ז', חובבי רובוטיקה, מורים במדעים או סתם סקרנים – הגיע הזמן להכיר מקרוב את אחד הכלים הכי פופולריים בתחום המייקרים והחינוך הטכנולוגי.

Arduino הוא שם של **פלטפורמה אלקטרונית בקוד פתוח** (Open Source) המבוססת על **לוח בקרה קטן** (לרוב בגודל של כרטיס אשראי) שמכיל מיקרו-בקרי – כלומר, **מוח קטן שמסוגל לקבל החלטות ולבצע פקודות**. לוח הארדואינו מתוכנת על ידי המחשב באמצעות **שפת תכנות פשוטה** ומתחבר לרכיבים כמו נורות, חיישנים, מנועים, כפתורים, 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**.

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

לוח Arduino כולל מספר רכיבים עיקריים:

- ## 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.

מי משתמש בארדואינו?

- תלמידים ומורים כמעט בכל תיכון טכנולוגי, מגמת פיזיקה או מדעי המחשב תמצאו היום ערכה עם לוחות Arduino.
- תלמידים כותבים קוד שמדליק נורות, מפעיל מנועים, מודד טמפרטורה ועוד – הכול בידיים שלהם.
- חוקרים ומהנדסים כן, גם באוניברסיטאות ומעבדות מחקר משתמשים בארדואינו – כפתרון מהיר, קל וזול לבדיקת רעיונות או אבות טיפוס.
- אמנים ויוצרים במוזיאונים, בפסטיבלים ובמיצבים – האינטראקטיביות שמביא איתו הארדואינו הפכה אותו לכלי פופולרי באמנות.
- חובבים (Makers) ארדואינו הוא לב ליבו של תרבות ה-Maker – אלו שיוצרים בבית רובוטים, תחנות מזג אוויר, מכשירים חכמים, גאדג'טים ועוד.

Tinkercad Circuits הוא סימולטור אונליין של ארדואינו – כלומר, אפשר "לבנות מעגלים" ולתכנת ארדואינו **בלי חומרה אמיתית**, הכול מהדפדפן!

יתרונות:

- אין צורך לקנות רכיבים פיזיים.

- אפשר להתנסות, לטעות ולתקן בלי לפחד "לשרוף" משהו.
- מעולה ללימוד בכיתה או מרחוק.

פרק ב': עבודה עם לוח חיבורים (Breadboard) ונורות LED

המשימות המרכזיות בפרק זה:

1. הבנה בסיסית

- הכרת מהו לוח חיבורים (Breadboard) ואיך הוא בנוי.
- ללמוד איך משתמשים בו להרכבת מעגלים בלי הלחמה.
- הבנת למה חשוב לחבר נכון את פסי המתח והאדמה.

2. עבודה מעשית

- חיבור לוח Arduino ללוח החיבורים בעזרת חוטי ג'אמפר.
- בניית מעגל פשוט עם לד ונגד כדי לבדוק את החיבור.
- ללמוד לזהות את הכיוונים של רכיבים (לד, דיודה וכו').

3. תרגול והתנסות

- הרכבת מעגל שמדליק ומכבה לד.
- התנסות בהחלפת רכיבים במעגל (לד בצבעים שונים, נגדים בערכים שונים).
- בדיקת מה קורה אם לא מחברים נגד, והסקת מסקנות.

4. איך להתקדם נכון

- עבודה בצורה מסודרת: קודם חיבורי מתח ואדמה, ורק אחר כך רכיבים נוספים.
- שימוש בחוטים צבעוניים כדי לא להתבלבל (אדום למתח, שחור לאדמה).
- שמירת המעגל ב-Tinkercard כי הוא יישמש אותך גם בפרק הבא.

מהו לוח חיבורים?

לוח חיבורים – או בשמו האנגלי – Breadboard הוא אחד הכלים הבסיסיים והחשובים ביותר בלימודי אלקטרוניקה ופרויקטים עם Arduino. זהו לוח פלסטיק עם חורים קטנים, המאפשרים חיבור רכיבים אלקטרוניים **בלי הלחמה**, כלומר – אין צורך להמיס בדיל או להשתמש בכלים מסובכים. כל מה שצריך זה לדחוף את הרכיב לתוך הלוח – והחיבור מתבצע.

איך הלוח בנוי?

לוח החיבורים מורכב ממספר שורות ועמודות של חורים, שמתחתיהם נמצאות **פסי מתכת**. החורים מסודרים כך:

- בצדדים יש לרוב שני פסי חשמל – אחד לספק חשמל (VCC) והשני לאדמה (GND).
- במרכז הלוח יש רווח – (gap) דרכו מכניסים מעגלים משולבים (ICs) או מיקרו-בקרים.
- כל עמודה של חמישה חורים מחוברת פנימית – כלומר, חיבור רכיב לשניים או יותר מהחורים באותה עמודה יגרום להם להיות מחוברים חשמלית.

למה זה כל כך שימושי?

1. אין צורך להלחים

הלוח מושלם עבור מתחילים וסטודנטים. במקום להלחים, פשוט מחברים – אפשר לנתק ולהחליף בקלות.

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 Arduino. אפשר לחבר את יציאות הלוח ישירות ללחם החיבורים באמצעות חוטים (Jumpers) ולשלוט בלדים, מנועים, חיישנים ועוד – ללא הלחמה וללא מעגלים קבועים.

פרק ג': שליטה בנורות LED באמצעות Scratch

המשימות המרכזיות בפרק זה:

1. הבנה תאורטית

- ללמוד איך לחבר בין סביבת **Scratch** לבין לוח **Arduino**
- הבנת מהו פלט דיגיטלי (ON/OFF) ואיך משתמשים בו לשליטה ברכיבים כמו לדים.
- הכרת מבנה התוכנה: פקודות **on start** לולאת **forever** פקודות הדלקה/כיבוי והשהייה.

2. עבודה מעשית

- התחברות לאתר **Tinkercad** ובניית מעגל עם לוח **Arduino** ולוח חיבורים (Breadboard)
- חיבור לד ונגד בצורה נכונה ותפעיל אותו תחילה ללא קוד, כדי לוודא שהחיבור תקין.
- העברת שליטת הלד לארדואינו וכתובת קוד פשוט בסקראץ המדליק ומכבה אותו.

3. תרגול והתנסות

- לשנות את הקוד כך שהלד יידלק ויכבה בקצב שונה (זמן השהייה אחר).
- התנסות בהוספת לדים נוספים וכתובת קוד לשליטה בהם לפי סדר.
- ביצוע ניסוי קטן: מה קורה אם מחליפים את הפינים הדיגיטליים או את ערך הנגד?

4. איך להתקדם נכון

- לעבוד מסודר: תחילה בדוק את המעגל בלי קוד, ואז עם קוד.
- להשתמש בהערות (Comments) בתוך Scratch כדי להבין את שלבי הקוד שלך.
- שמירת הקובץ ואת המעגל ב-Tinkercad בכדי להשתמש בהם גם בפרקים הבאים.

איך לחבר את **Scratch** ל-**Arduino**

היכנסו לאתר www.tinkercad.com

בפעם הראשונה לוחצים על פקד "Sign Up" בצד ימין עליון של המסך:



עברו למסך הבא:

Start Tinkering

How will you use Tinkercad?

In school?

Educators start here

Students, join a Class

On your own

Create a personal account

Already have an account?

[Sign In](#)

קבלו את קוד הכיתה מהמורה והתחברו (או שאפשר להתחבר גם כ- Personal Account).

Join Class

Type the code your teacher shared

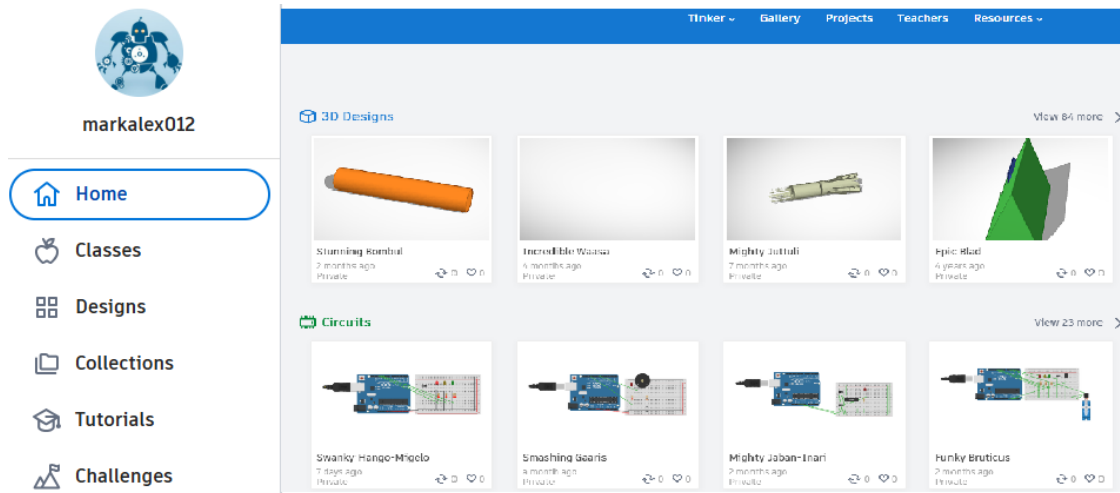
Like: 123 456 789

Go to my class

Not Joining a class?

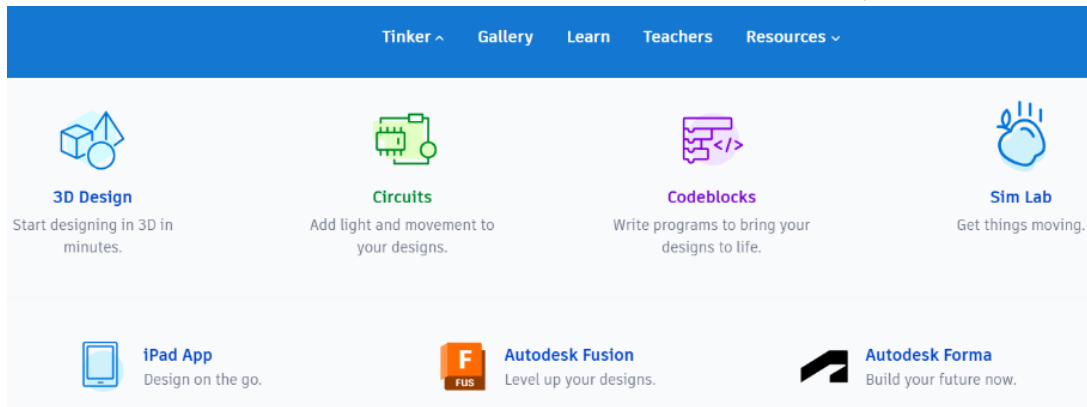
[Go Back](#)

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 Codeblocks, Circuits, 3D Design 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.

10

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.

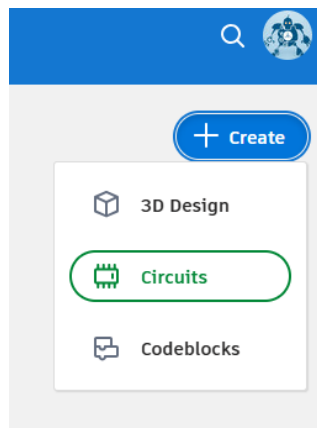
Sim Lab ביצוע סימולציות והדמיות תנועה ואינטראקציות פיזיקליות.

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

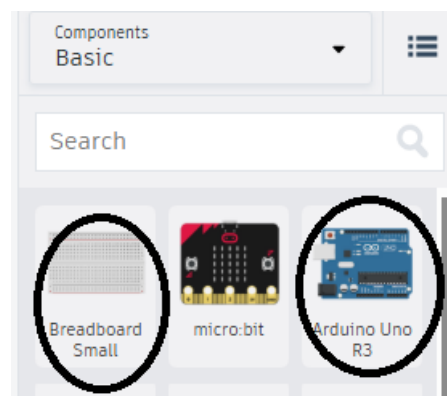
Autodesk Fusion כלי מקצועי לעיצוב והנדסה מדויקת של מוצרים.

Autodesk Forma כלי לתכנון סביבתי ואדריכלות חכמה.

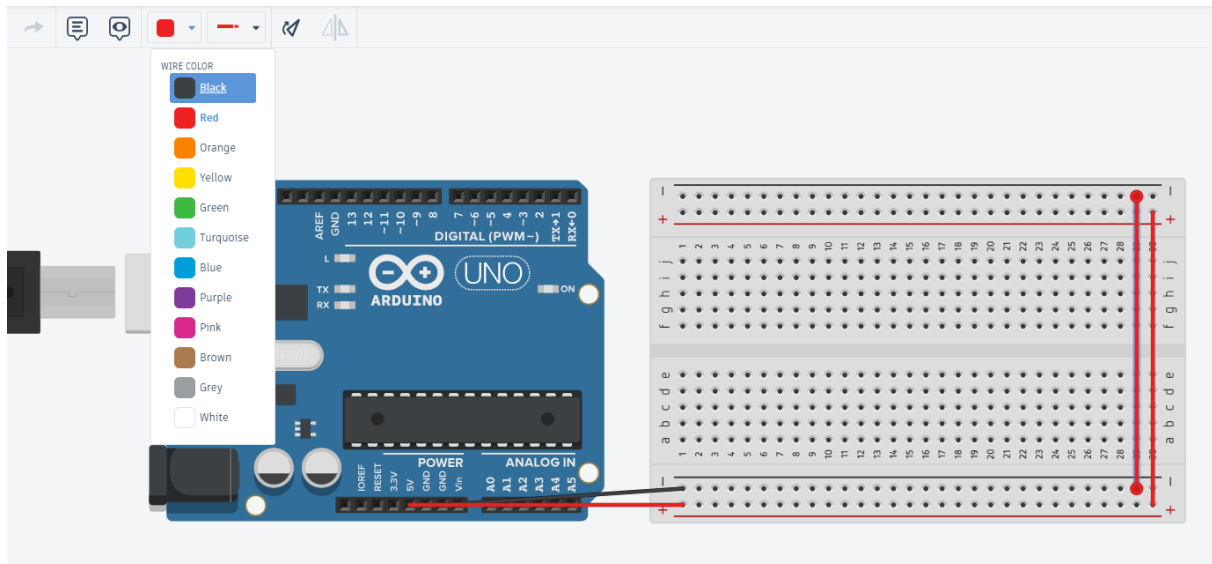
היכנסו לחשבון בצורה הבאה:



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



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

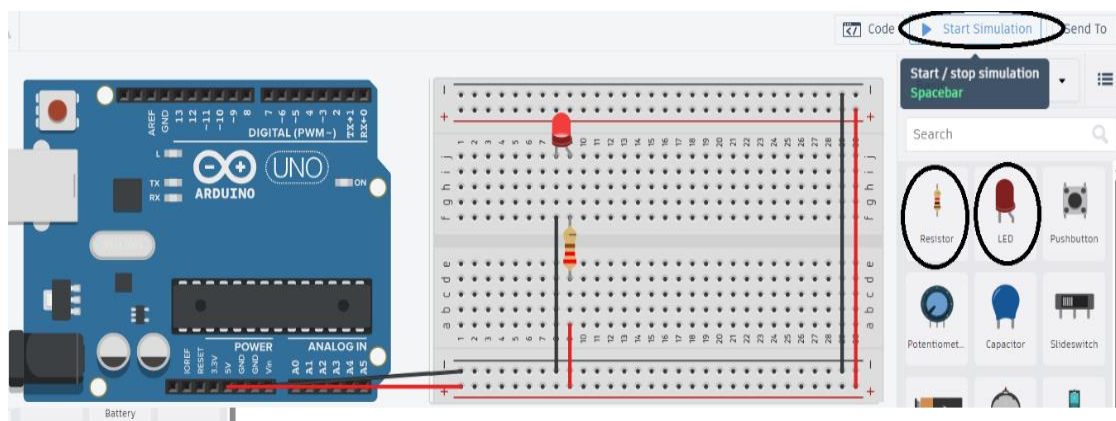


שימוש בפלטים דיגיטליים

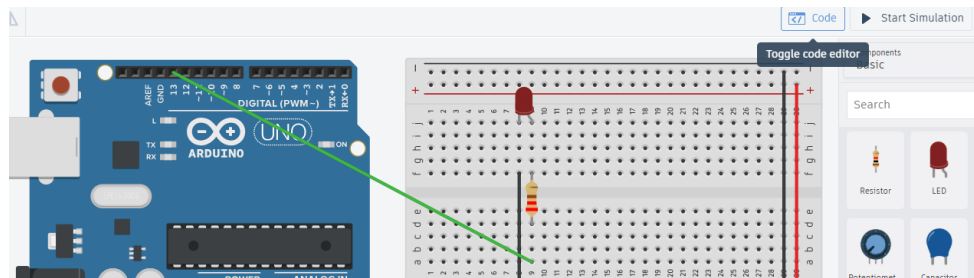
חברו לד והפעילו הדלקה וכיבוי לד.

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.

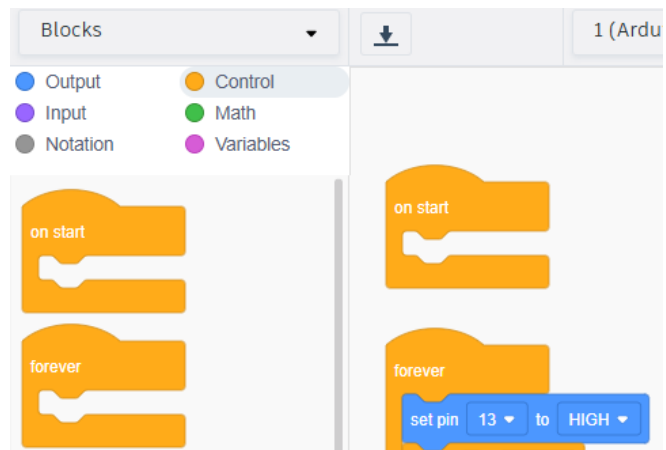


העבירו רגל הפעלת לד לאחד הפורטים, נניח לפורט 13 והפעילו קוד בשפת Scratch.



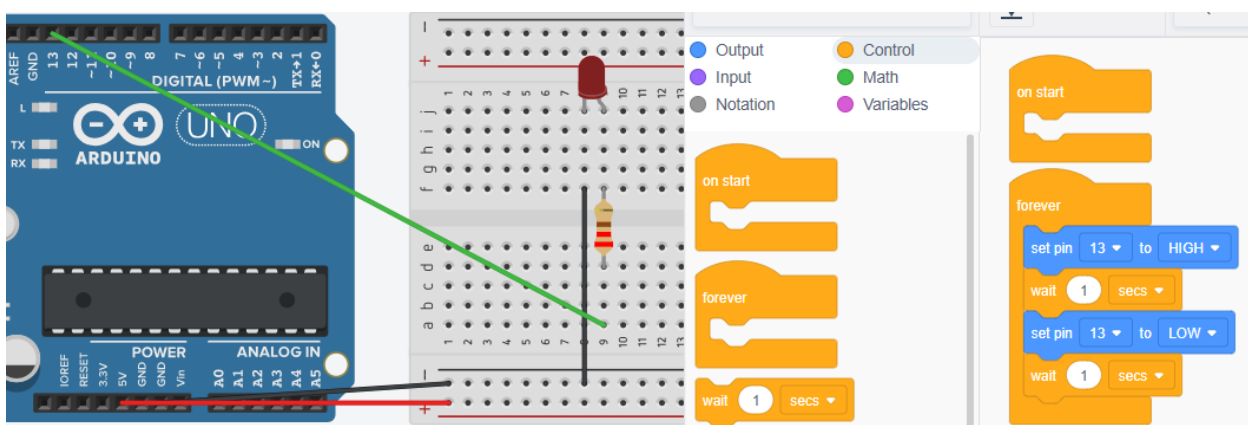
עברו לקוד בעזרת כפתור **Code**.

מתפריט Control בחרו פונקציה on start ופונקציה (לולאה) forever.



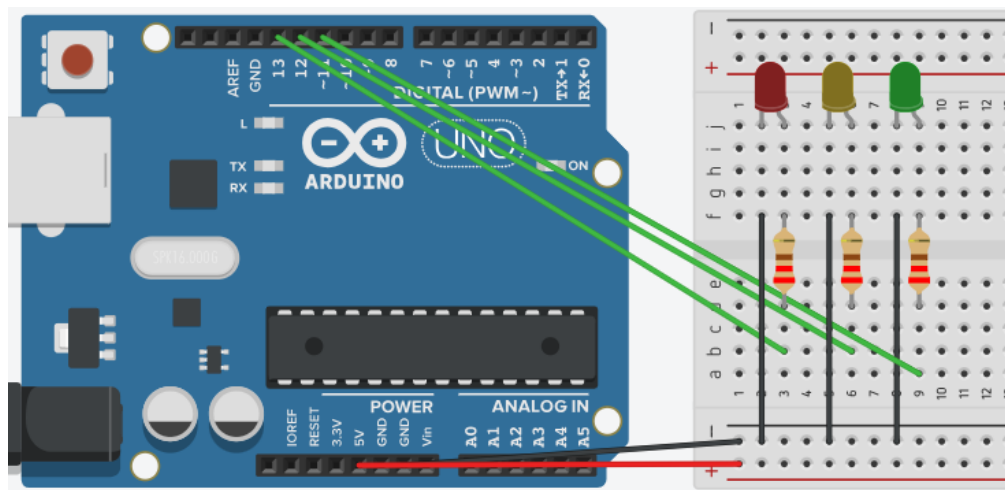
פונקציה on start מכילה הגדרות והפעלות חד פעמיות ולולאה forever שפועלת בצורה אינסופית.

בינתיים השאירו פונקציה on start ריק והוסיפו הדלקת וכיבוי לד מתפריט Output (צבע כחול) והשהייה wait מתפריט Control.



תרגיל בית: "רמזור אוטומטי".

בנו רמזור בצורה הבאה. חברו 3 לדים: אדום, צהוב, ירוק ומפעילים אותם בצורה הבאה:
 הפעלת צבע אדום במשך 3 שניות.
 הפעלת 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.
 המעגל נראה בצורה הבאה:



פרק ד'. עבודה עם כפתורים ומתגים

המשימות המרכזיות בפרק זה:

1. הבנה תאורטית

- הכרת שני סוגי הכפתורים הנפוצים ב-Tinkercad (לחצן רגעי) ו-Slide Switch מתג שנשאר במצב שהוגדר.
- ללמוד מהו קלט דיגיטלי ואיך ניתן להשתמש בו כדי לשלוט במעגלים.
- הבנת עיקרון הכפתור שולח ערך "0" או "1", והקוד מחליט איך לפעול בהתאם.

2. עבודה מעשית

- חיבור כפתור או מתג ללוח החיבורים (Breadboard) ולארדואינו.
- כתיבת קוד ב-Scratch שמדליק ומכבה לד לפי מצב הכפתור.
- ללמוד איך המערכת מתנהגת בעת לחיצה, שחרור, או שינוי את מצב המתג.

3. תרגול והתנסות

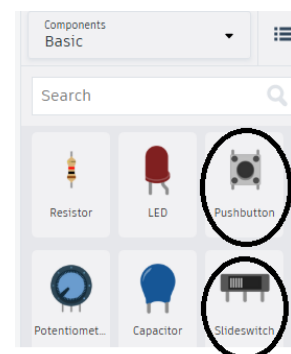
- לשנות את הקוד כך שהלד יבהב במקום להידלק רצוף.
- התנסות בשילוב שני כפתורים (למשל: אחד מדליק, אחד מכבה).
- ביצוע ניסוי: מה קורה אם מחברים את הכפתור בלי נגד משיכה (Pull-down resistor)?

4. איך להתקדם נכון

- לעבוד מסודר: קודם בדיקת המעגל בחיבורים נכונים, ורק אחר כך הוספת קוד.
- לשים לב לסימן "=" בקוד – זהו תנאי שמשווה ערך (ולא מגדיר אותו).
- שמירת המעגל והקוד, כי בהמשך נשתמש בכפתורים גם לפרויקטים מורכבים יותר (כמו רמזור עם שליטה).

איך עובד כפתור?

ישנם 2 סוגי כפתורים ב-Tinkercad:

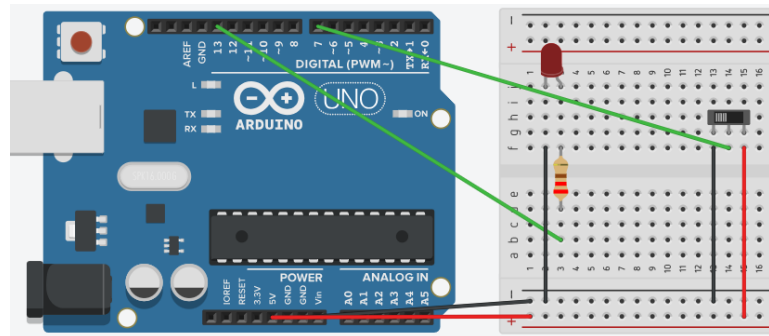


ההבדל ביניהם:

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.

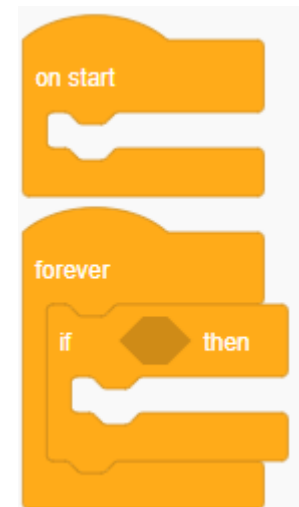
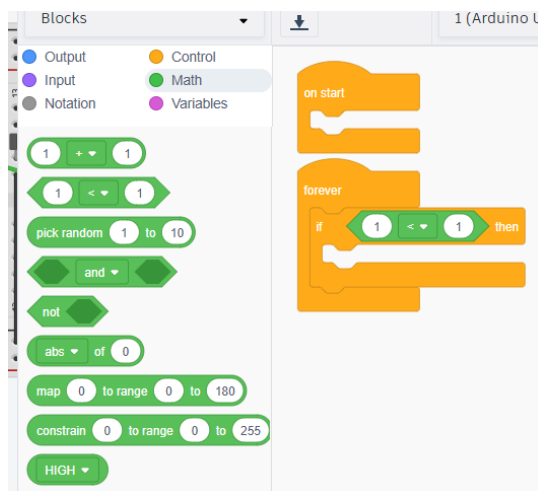
שליטה בפעולות דרך כפתור ב-Scratch

חברו קודם Slideswitch בצורה הבאה:



כתבו קוד חדש:

הכניסו תנאי if מתוך תפריט Control ללולאה: הוסיפו תנאי מתוך תפריט Math:

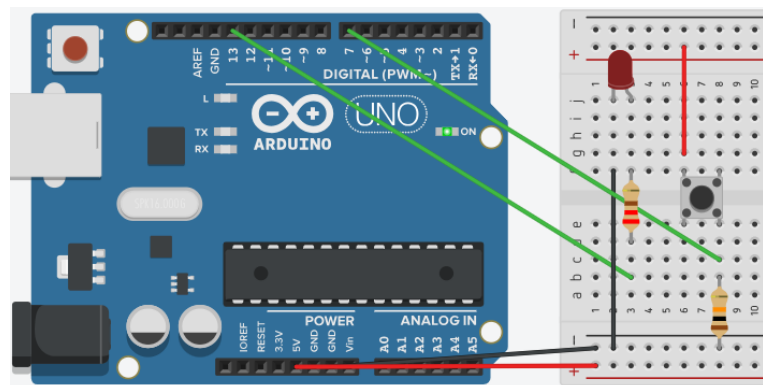


עברו לתפריט Input ובחרו read digital pin והוסיפו קוד של הדלקת וכיבוי של לד. שימו לב על שינוי תנאי – סימן " $=$ ".



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

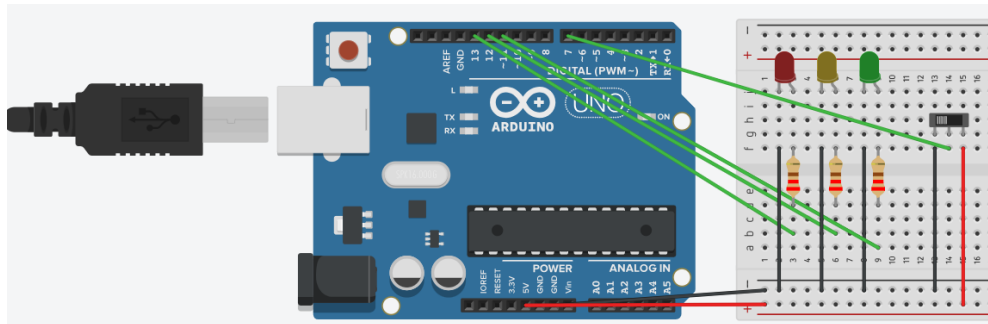
שימוש בכפתור לחיצה



כל עוד לא לוחצים על הכפתור – לד יהיה כבוי, אם תלחצו ותחזיקו את הכפתור – לד מהבהב.

קוד נשאר אותו קוד כמו בדוגמה עם Slideswitch.

תרגיל בית: "רמזור אוטומטי עם שליטה מכפתור"



אם Slideswitch נמצא במצב "1" – רמזור עובד כרגיל (כמו שהוגדר בשיעורי בית הקודמים), אם תעבירו כפתור למצב "0", רמזור עובר למצב "שבת" (אור צהוב מהבהב אם השהייה קבוע של 0.5 שניות הפעלה, 0.5 שניות כיבוי).

פרק ה': משתנים ואופרטורים לוגיים ב-Scratch + Arduino

המשימות המרכזיות בפרק זה

1. הבנה תאורטית

- ללמוד מהו **משתנה**: כלי לשמירת מידע (כמו מונה או מצב).
- הבנת מה הם **אופרטורים לוגיים** ($>$, $<$, $=$) ואיך מאפשרים לקבל החלטות בקוד.
- הכרת השילוב בין משתנים ותנאים כדי לשלוט בפעולות בצורה חכמה יותר.

2. עבודה מעשית

- יצירת משתנה חדש ב-Scratch ושימוש בו לשמירת מידע מהמעגל.
- חיבור כפתור ללוח Arduino וכתובת קוד שבו כל לחיצה מגדילה את ערך המשתנה.
- הוספת תנאי לוגי (if) שמפעיל לד או רכיב אחר רק כשהערך של המשתנה עובר סף מסוים.

3. תרגול והתנסות

- התנסות בשינוי התנאים (לדוגמה: מה יקרה אם נדרוש שהערך יהיה גדול מ-5?).
- התנסות בהוספת אופרטורים מורכבים יותר (and, or, not) כדי ליצור לוגיקה מתקדמת.
- ביצוע פרויקט קטן "**ספירת לחיצות**" מונה שמציג את מספר הלחיצות במסך הסריאלי ומפעיל לד בהתאם.

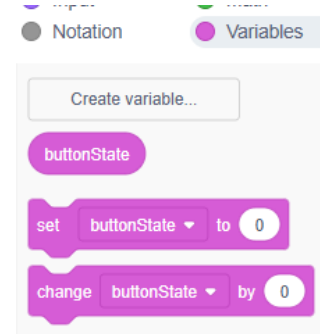
4. איך להתקדם נכון

- בדקו בכל שלב את ערך המשתנה והדפיסו במסך או הציגו בסקראץ'.
- עבדו מסודר – קודם חיבור נכון של המעגל, אחר כך כתיבת הקוד.

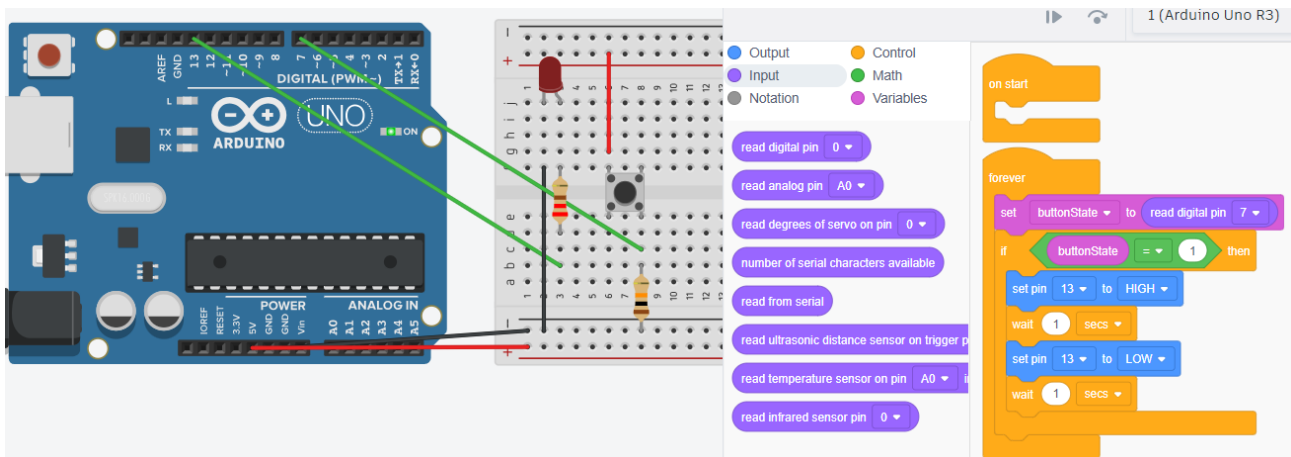
- שמרו את המעגל והקוד ב-Tinkercad כי בהמשך תשתמשו במשתנים גם בפרויקטים מתקדמים יותר (כמו בקרה על חיישנים).

איך לשמור מידע באמצעות משתנים?

ניתן להגדיר משתנים כדי לשמור מצבים ומספרים בתפריט Variables.



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



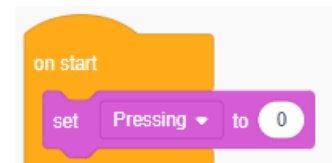
שימוש בתנאים לשליטה בפעולות

בדקו את השילוב תנאים בסיסיים ביותר לשילוב תנאים.

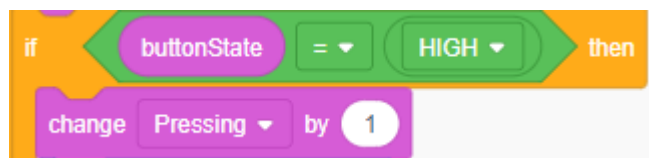


פרויקט: "ספירת לחיצות"

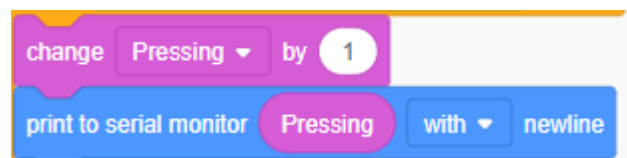
צרו משתנה Pressing ואפסו אותו בכניסה לתוכנית.



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



הדפיסו ערך של ספירה במסך סריאלי:



תרגיל בית: הצגת ערך של משתנה

קחו תרגיל אחרון עם רמזור ומפסק. הגדרו משתנה "State" אם מפסק נמצא במצב "1" משתנה יכיל ערך "1", אחרת משתנה יכיל ערך "0". אם משתנה מכיל ערך "1" הציגו על מסך סריאלי מילה "True" אחרת הציגו מילה "False".

פרק ו': שימוש באותות אנלוגיים

משימות מרכזיות בפרק זה

1. הבנה תאורטית

- ללמוד מה ההבדל בין אות דיגיטלי (0/1) לבין אות אנלוגי (ערכים רציפים).
- הבנת מהו ממיר (ADC (Analog to Digital Converter ואיך הוא מאפשר לארדואינו "לקרוא" ערכים משתנים.
- הכרת מושגים כמו רזולוציה 10 ביט ב-Arduino UNO) וטווח קריאה (0–1023)

2. עבודה מעשית

- חיבור פוטנציומטר לארדואינו וביצוע קריאה של ערכים אנלוגיים בפורט A0.
- רשימת קוד ב-Scratch שמציג את הערך הנמדד בזמן אמת.
- התנסות בבניית מחלק מתח עם חיישן אור (LDR) ובקריאת ערכי האור.

3. תרגול והתנסות

- ביצוע מדידות מתח אמיתיות והשוואה לערכים שהתקבלו במסך הסריאלי.
- בניית טבלה של ערכים (מתח ↔ ערך דיגיטלי) וניתוח את התוצאות.
- ביצוע ניסוי: מה קורה אם משנים את ערך הנגד במחלק המתח?

4. איך להתקדם נכון

- לבדוק תמיד את החיבורים לפני הפעלה, כדי למנוע שגיאות.
- לעבוד עם חוטים קצרים ומסודרים כדי לצמצם רעשים חשמליים.
- שמירת המעגלים ב-Tinkercad והכנה לשלב הבא – שימוש בחיישנים לשליטה על רכיבים אחרים (כמו נורות לד).

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.

בואו נבין איך זה עובד, ונתרגל גם חישוב רזולוציה והמרת מתח לערך דיגיטלי.

מה זה אות אנלוגי?

אות אנלוגי הוא אות רציף – לדוגמה:

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

אבל מחשב וגם Arduino מבינים רק מספרים בינאריים – ולכן יש צורך להמיר את המתח האנלוגי לערך דיגיטלי. כאן נכנס לפעולה ה-ADC.

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

איך ה-ADC עובד ב-Arduino?

בלוחות כמו Arduino UNO יש ADC ברזולוציה של 10 ביט זה אומר:

- הוא מחלק את טווח המתחים (0–5V) ל- 2^{10} 1024 רמות שונות.
- כל רמה מיוצגת במספר בין 0 ל-1023.

כך:

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

מה משפיע על הדיוק?

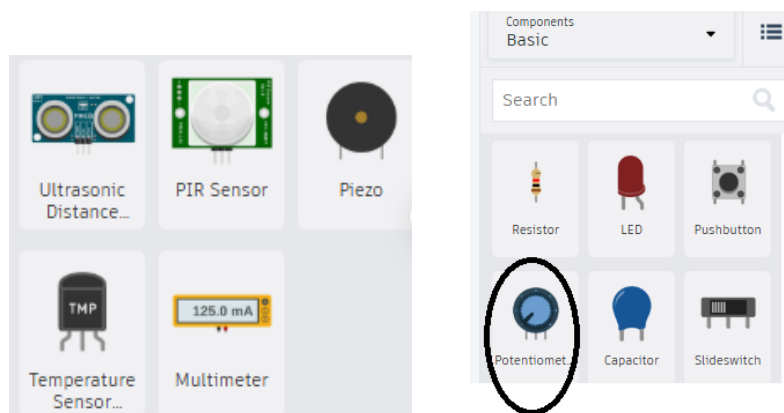
1. מתח ייחוס לא מדויק: ב-UNO ברירת המחדל היא 5V, אבל במציאות זה עלול להיות 4.9V או 5.1V.
2. רעשים חשמליים: חוטים ארוכים, מקורות מתח לא יציבים וכו'.
3. חיישנים לא ליניאריים: ייתכן שהחיישן לא מספק מתח מדויק יחסית לערך הנמדד.
4. טווח קריאה: ניתן להגדיר Vref אחר, לדוגמה 3.3V או 1.1V, כדי לקבל רזולוציה גבוהה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

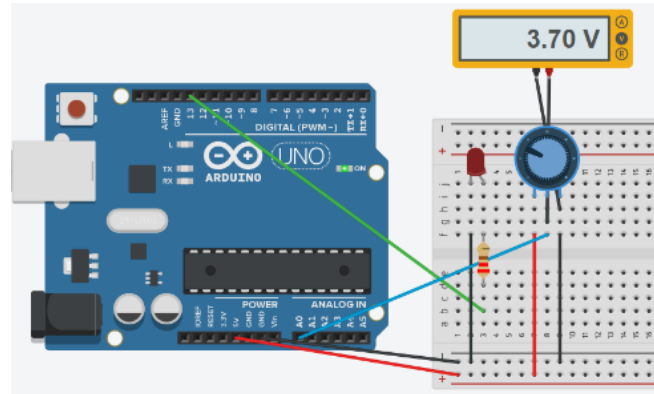
חיבור פוטנציומטר לפורט אנלוגי

בחרו נגד משתנה מתפריט בסיסי.

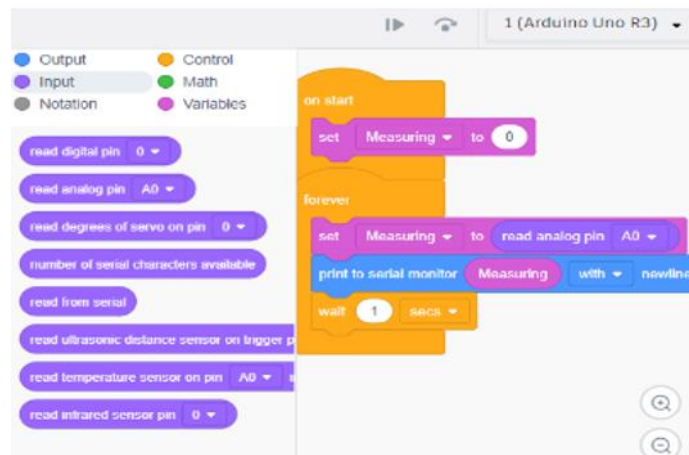


22

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

מידת 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?

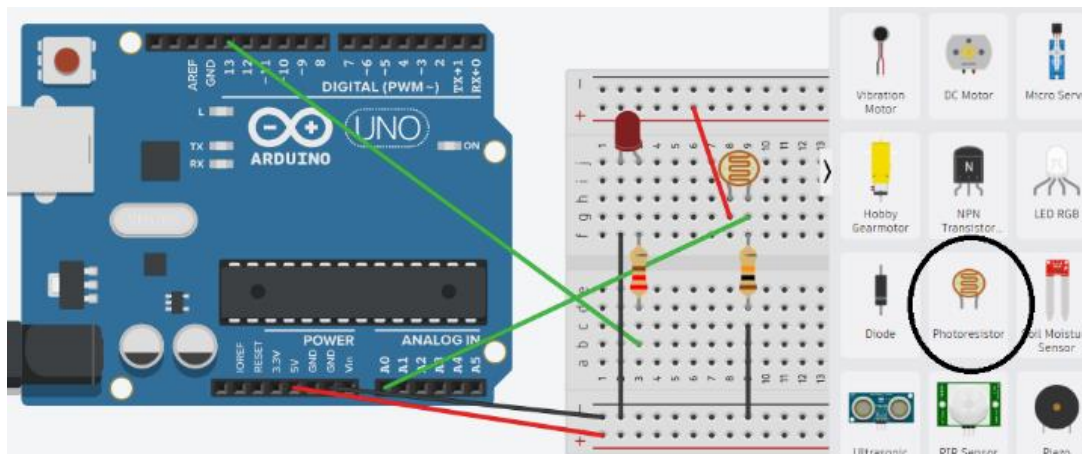
חיישן אור – (Light Sensor) ובעברית: חיישן תאורה – הוא רכיב שמודד את עוצמת האור בסביבתו. החיישן הנפוץ ביותר בלמידה בסיסית הוא LDR ראשי תיבות של Light Dependent Resistor כלומר: נגד שתלוי בעוצמת האור.

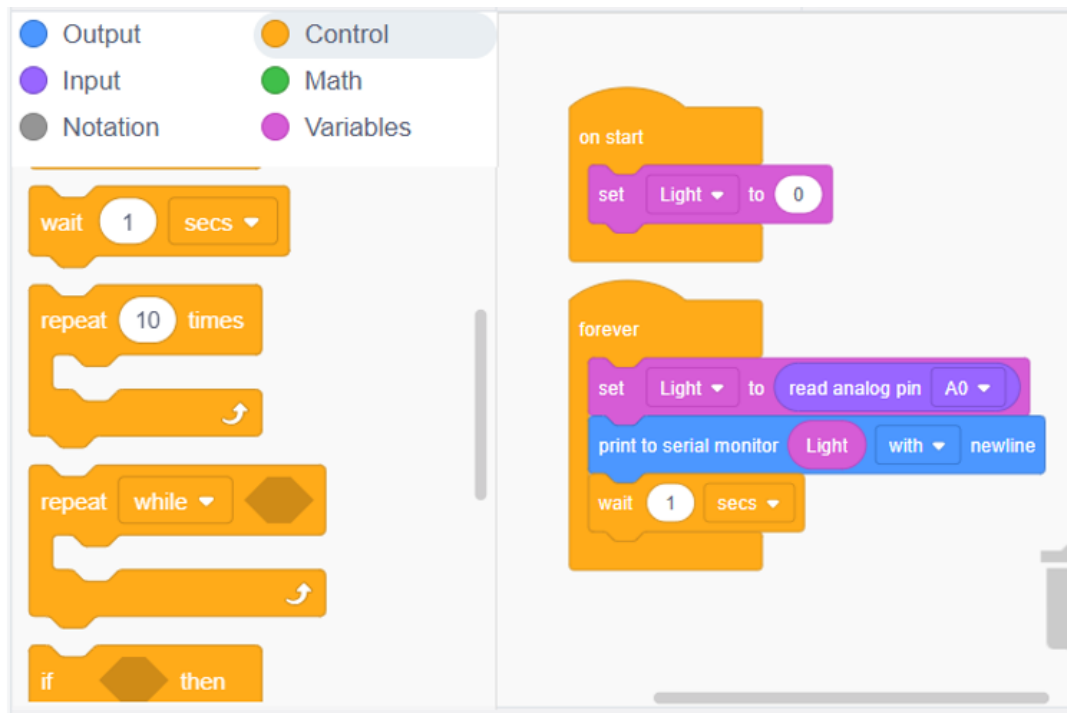
- כשיש הרבה אור, ההתנגדות של ה-LDR נמוכה.
- כשחשוך, ההתנגדות גבוהה.

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().



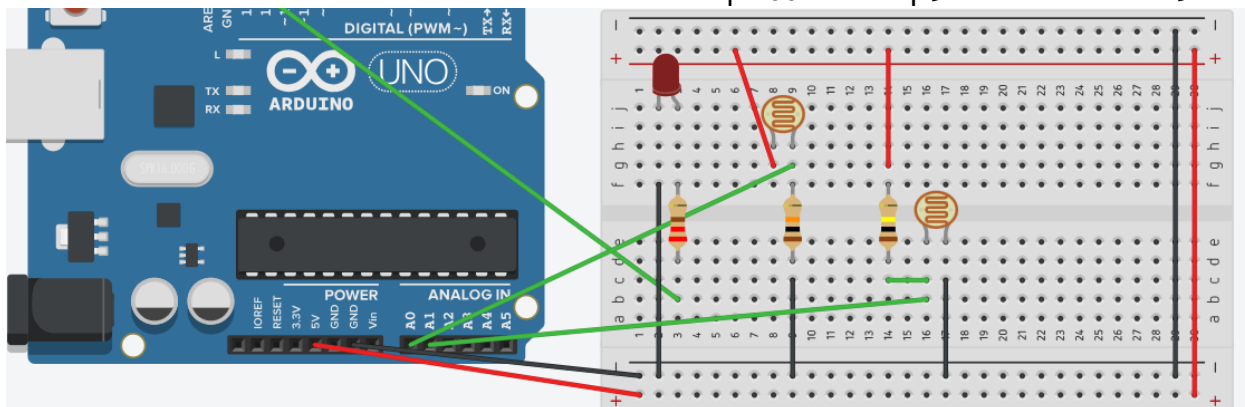


שיעורי בית. שליטה בנורות לפי רמת האור

בנו פרויקט חדש שמודד רמת האור ומדליק 3 לדים בהתאם לתנאים הבאים: לד אדום נדלק אם אור הסביבה חזק, לד ירוק נדלק אם רמת האור בינונית ולד כחול נדלק אם רמת האור נמוכה.

שימו לב:

חיבור חיישן אור בצורה ישירה ובצורה הפוכה.
 שינוי ערכים בהתאם לשינוי ערך הנגד ל- 100 קילו אוהם.



פרק ז': שימוש בזמזם (פיזואלקטרי) ליצירת צלילים

משימות מרכזיות בפרק זה

1. הבנה תאורטית

- הכרת שני סוגי הזמזמים **Active Buzzer** (שמפיק צליל קבוע) ו- **Passive Buzzer** שמאפשר שליטה בתדר ובמנגינות.
- הבנת איך אפשר לייצר צלילים שונים על ידי שליחת תדרים שונים לארדואינו.
- ללמוד מה ההבדל בין תדר נמוך לתדר גבוה, ואיך זה משפיע על הצליל שנשמע.

2. עבודה מעשית

- חיבור זמזם ללוח Arduino אנודה לפין דיגיטלי, קתודה ל-GND.
- כתיבת קוד ב- Scratch שמפיק צפצוף פשוט.
- התנסות בשינוי התדרים וכיצד הצליל משתנה בהתאם.

3. תרגול והתנסות

- התנסות ביצירת צלילים בקצב שונה (מהיר/איטי) על ידי שילוב פקודות השהייה.
- התנסות בהפקת "מנגינה" קצרה באמצעות סדרת תדרים שונים.
- ביצוע ניסוי: מה קורה אם מחליפים את הפין הדיגיטלי או מוסיפים נגד בסדרה עם הזמזם?

4. איך להתקדם נכון

- לעבוד מסודר: קודם לוודא חיבורים נכונים, אחר כך להריץ קוד.
- כתיבת קוד בשלבים – להתחיל מצפצוף פשוט, ואחר כך להוסיף שינויים ותדרים שונים.
- שמירת המעגל ואת הקוד, בכדי להשתמש בזמזם גם בפרויקטים מתקדמים יותר (למשל אזעקות או שילוב מנגינה בפרויקט רובוטי).

איך מפיקים צלילים עם זמזם?

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

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

מה זה Buzzer?

ישנם שני סוגי זמזמים עיקריים:

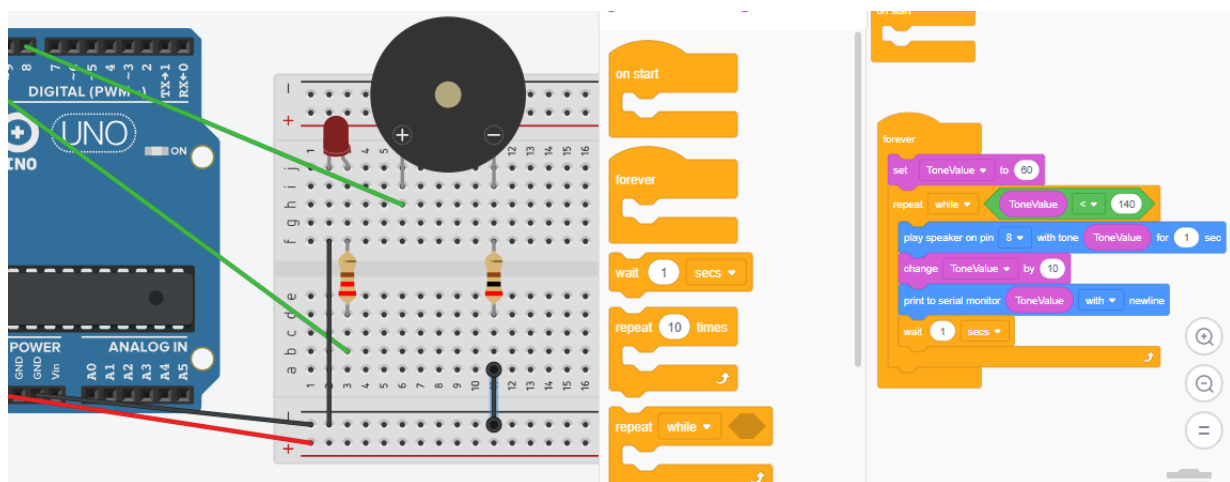
1. **Active Buzzer** משמיע צליל קבוע כשהוא מקבל זרם (כמו כפתור של מיקרוגל). לא ניתן לשלוט על התדר.

2. **Passive Buzzer** רכיב שמשיע צליל בהתאם לתדר שאנחנו שולחים אליו. זה מאפשר הפקת מנגינות ואפקטים. בכדי לגנן מנגינה (ולא רק צפצוף פשוט) יש להשתמש ב-Passive Buzzer.

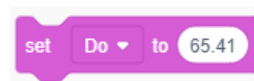
איך מחברים את הזמזם?

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

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



שיעורי בית: יצירת מנגינות ב-Scratch
 מצאו טבלת תדרים של תווי מוזיקה.



שייכו שם המשתנה לתדר הרצוי. לדוגמה:



מצאו מנגינה קצרה (שימו לב שבחלק מחומרים משתמשים בתווים Do, Re, בחלק בציור גרפי: A – La, C – Do, ובחלק מסימנים בין לאומיים של תווים: C – Do, כתבו קוד להפעלת שיר. במידה והכל תקין שמרו מעגל וקוד.

פרק ח': PWM ומנועי סרוו (תנועה מכנית)

משימות מרכזיות בפרק זה

1. הבנה תאורטית

- הבנת מה זה – **PWM (Pulse Width Modulation)** טכניקה שמאפשרת לשלוט בעוצמה ממוצעת של מתח/זרם על ידי הדלקה וכיבוי מהירים מאוד.
- ללמוד את המושג **Duty Cycle** ואיך שינוי שלו משנה את עוצמת האור בלד או את מהירות מנוע.
- הכרת מהו **מנוע סרוו**, במה הוא שונה ממנוע רגיל, ואיך את PWM קובע את הזווית שבה הוא נעצר.

2. עבודה מעשית

- חיבור לד לפין PWM ושליטה על הבהירות שלו באמצעות הפקודה **analogWrite()**
- חיבור למנוע סרוו פשוט למשל (SG90) וכתובת קוד שמזיז אותו לזוויות שונות. (0° , 90° , 180°)
- התנסות בשינוי ערכי ה-PWM והשפעת שינוי זה על רכיבים שונים.

3. תרגול והתנסות

- ביצוע פרויקט קטן: שליטה הדרגתית על בהירות לד (עמעום).
- התנסות בהפעלת מנוע סרוו כך שישמש כ"מחסום" שנפתח ונסגר לפי תנאי בקוד.
- ביצוע ניסוי: מה קורה אם משנים את תדר ה-PWM או משתמשים בפין שאינו תומך ב-PWM?

4. איך להתקדם נכון

- לעבוד מסודר: לזהות את הפינים התומכים ב-PWM מסומנים ב-~
- לבדוק בכל שלב שהמעגלים מחוברים נכון כדי למנוע קצר.
- לשמור את הקוד והמעגלים, בכדי להשתמש ב-PWM ובסרוו גם בפרויקטים מורכבים יותר (כמו רובוטיקה או אוטומציה).

בעולם האלקטרוניקה, המתח שיוספק 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 PWM 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-PMW, והוא אחוז הזמן שבו האות נמצא במצב HIGH (דלוק) בכל מחזור.

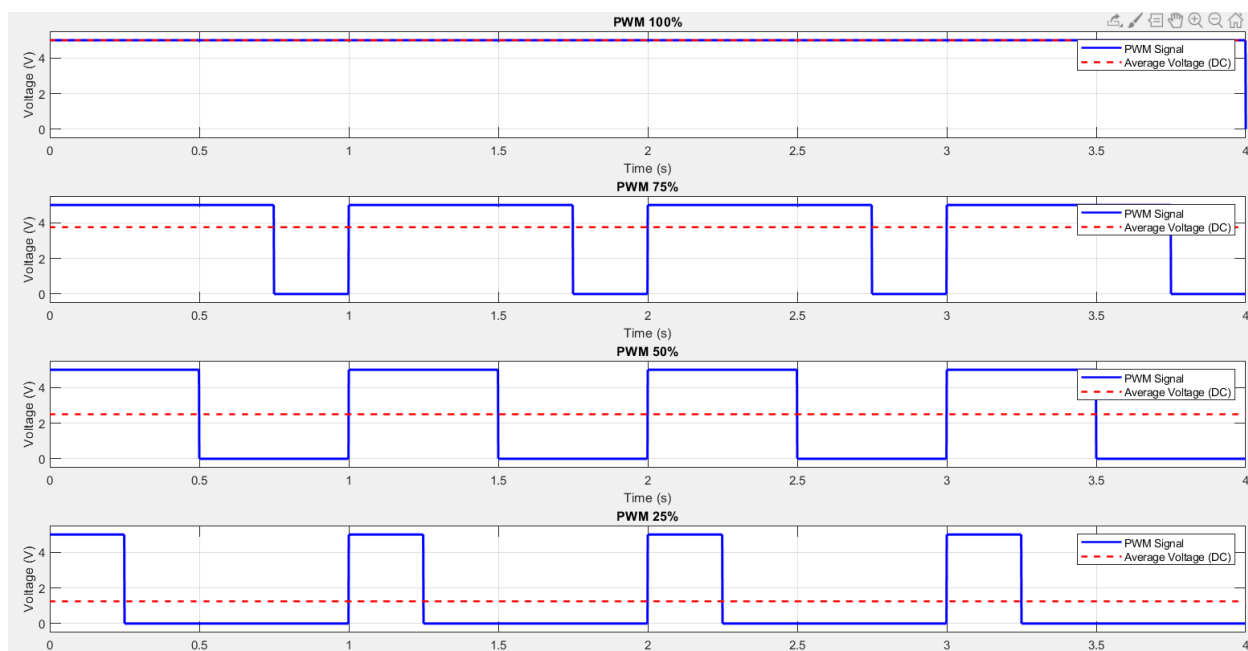
לדוגמה:

- 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.

PMW לעומת אות אנלוגי אמיתי

מאפיין	PMW	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% 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 גו.

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 ממנוע 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.

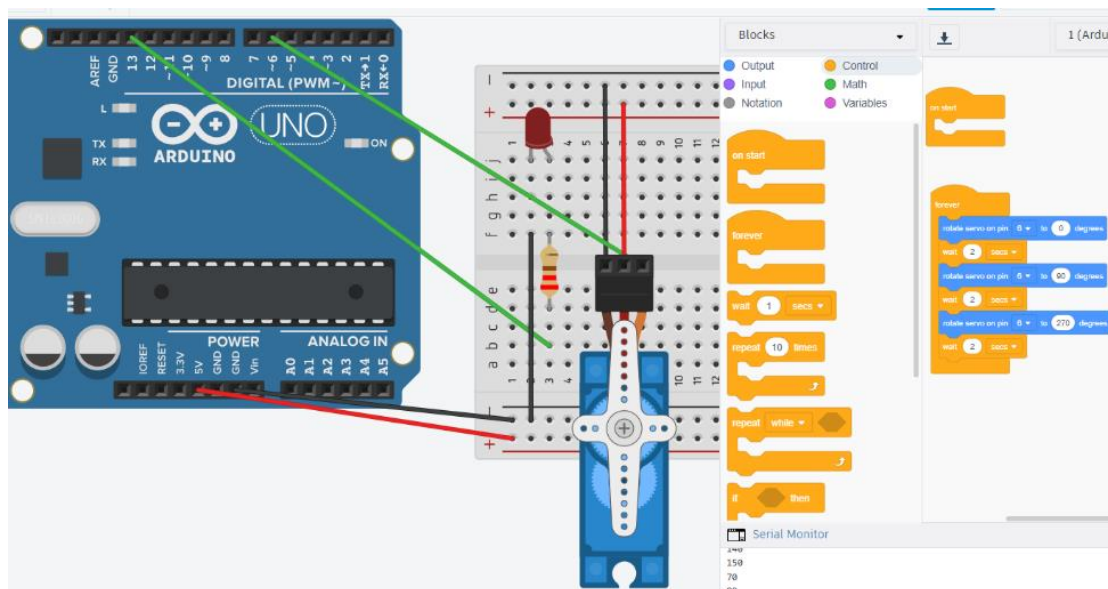
- בקר פנימי משווה בין הזווית הרצויה (שמגיעה מהארדואינו) לבין הזווית האמיתית, ואם יש הבדל – הוא מזיז את המנוע עד שהסרוו מגיע למקום הנכון.

איך שולטים בו?

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

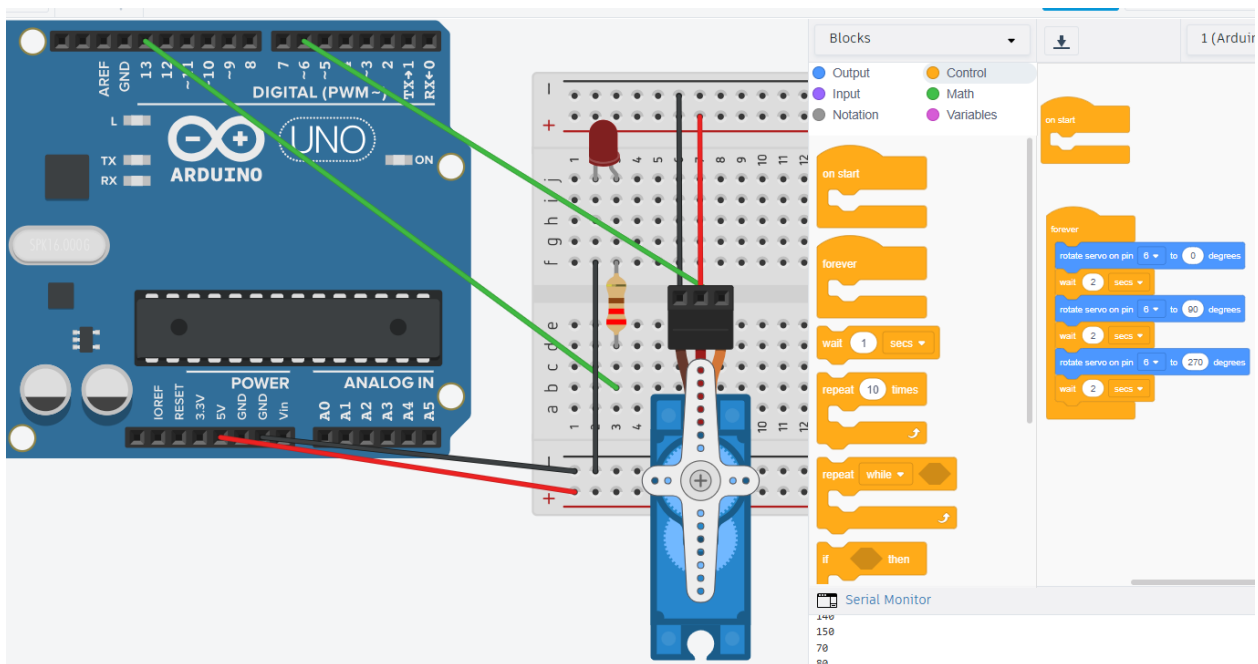
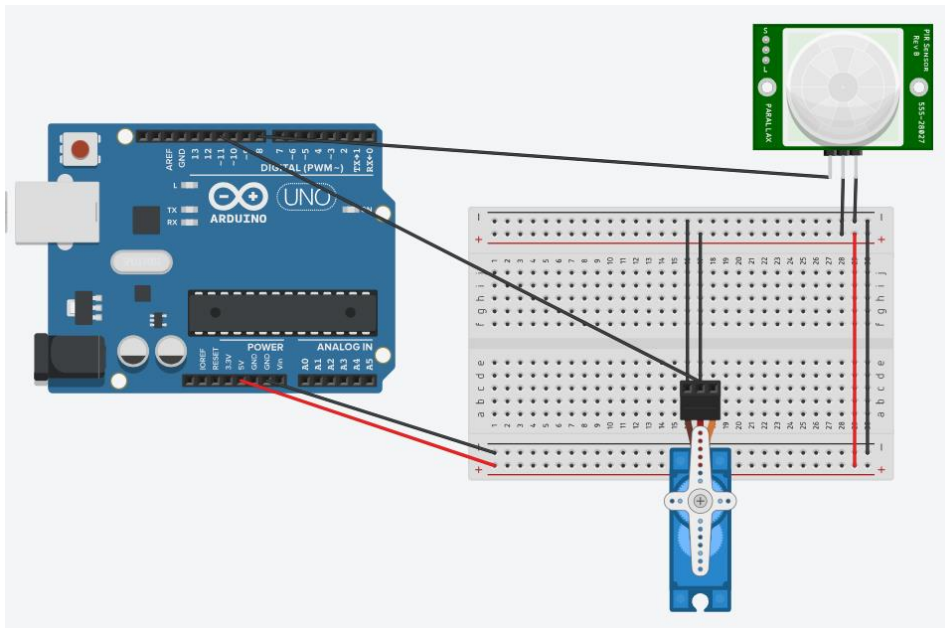
- $\sim 1\text{ms} \rightarrow 0^\circ$
- $\sim 1.5\text{ms} \rightarrow 90^\circ$
- $\sim 2\text{ms} \rightarrow 180^\circ$

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



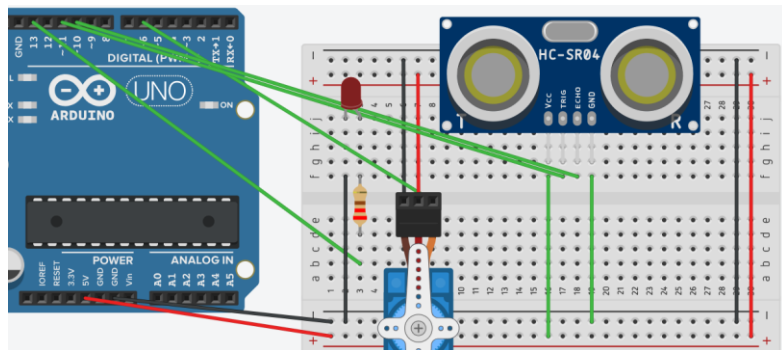
פרויקטון קטן:

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

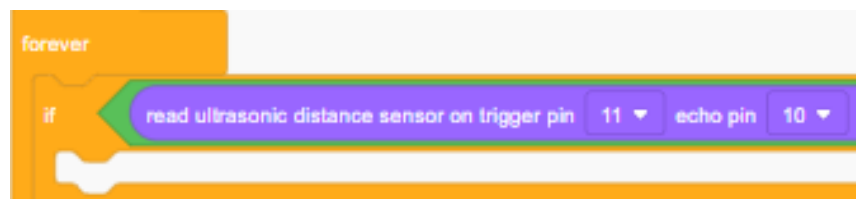


תרגיל בית: "מחסום חשמלי"

מטרת המעגל – אם מישור מתקרב למחסום במרחק מסוים, מחסום נפתח ואז לאחר זמן מסוים נסגר בעזרת מנוע סרוו. חברו חיישן מרחק ומנוע סרוו בצורה הבאה:



השתמשו בפקודה הבאה לקריאת מרחק:



סיכום

חוברת זו נבנתה עבור תלמידי כיתה י' במגמת הנדסה ישומית, והיא מהווה מדריך מעשי ולימודי להיכרות עם עולם ה-**Arduino** בשילוב סימולטור **Tinkercad** ושפת **Scratch**. מטרת החוברת היא לחשוף את התלמידים לעקרונות בסיסיים באלקטרוניקה ובתכנות בצורה פשוטה, חווייתית ונגישה, המשלבת בין תיאוריה ליישום בפועל.

במהלך הלמידה, התלמידים מתנסים בשלבים מדורגים: החל מהיכרות עם סביבת העבודה והבנת רכיבים בסיסיים (לוח חיבורים, לדים, כפתורים), דרך בנייה והפעלה של מעגלים פשוטים, ועד לשליטה בחיישנים, זמזמים ומנועי סרוו. כל פרק בנוי כך שהתלמיד מקבל הסבר תאורטי קצר, מתרגל עבודה מעשית בסימולטור, ומסיים במשימות ואתגרים שמחזקים את הידע ומעודדים יצירתיות.

מעבר לרכישת ידע טכנולוגי, החוברת שמה דגש על פיתוח **חשיבה לוגית, פתרון בעיות ועבודת חקר עצמאית**. באמצעות החיבורים בין Scratch (תכנות בבלוקים) לארדואינו (בקר פיזי), התלמידים מגלים איך רעיון מחשבתי יכול להפוך במהירות לפרויקט אמיתי שפועל מול עיניהם. חוברת זו מהווה בסיס ללימודים מתקדמים יותר בתחום ההנדסה, האוטומציה והרובוטיקה, ומקנה לתלמידים מיומנויות חיוניות לעולם הטכנולוגי של המחר.