

סדנה מעשית – ארדואינו – חלקים נבחרים

Copyright © 2025 Eldad Hellerman

All rights reserved. This document or parts thereof may not be reproduced in any form, stored in any retrieval system, or transmitted in any form by any means - electronic, mechanical, photocopy, recording, or otherwise - without prior written permission of the author.

You may contact me at EldadHellerman@gmail.com.

סילבוס

מספר מעבדה ודרגת קושי	A שימוש בספריות	B מימוש עצמאי של דברים	C הרחבות למתקדמים
0	מבוא, התקנת IDE. להעלות הבהוב ולראות שהכל עובד.		
1	שימוש בנורת לד.	שימוש במשתנים ו-DEFINEים.	הבהוב על בסיס חותמות זמן.
2	שימוש בכפתורים.		קריאת דף נתונים על נגדי פולאפ והסתכלות בסכמה כניסה בשבב.
5	זמזם פסיבי, נגינת מלודיה.		
12	משחק הזיכרון סיימון.		

חיבורי החומרה הנוכחית

Pin	Label name
D12	SW4
~D11	SW1
~D10	LED1
~D9	BUZ
D7	SW2
~D6	LED2
~D5	LED4
D4	SW3
~D3	LED3

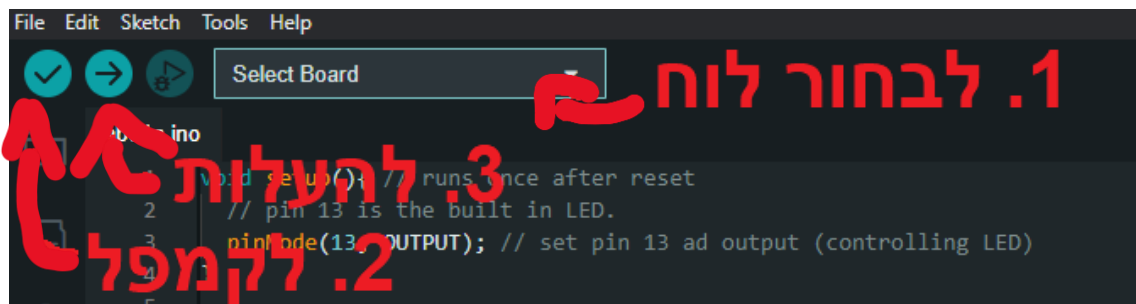
הרצאה 0

ניתן לקרוא על כל הפונקציות השונות של ארדואינו באתר הרשמי:

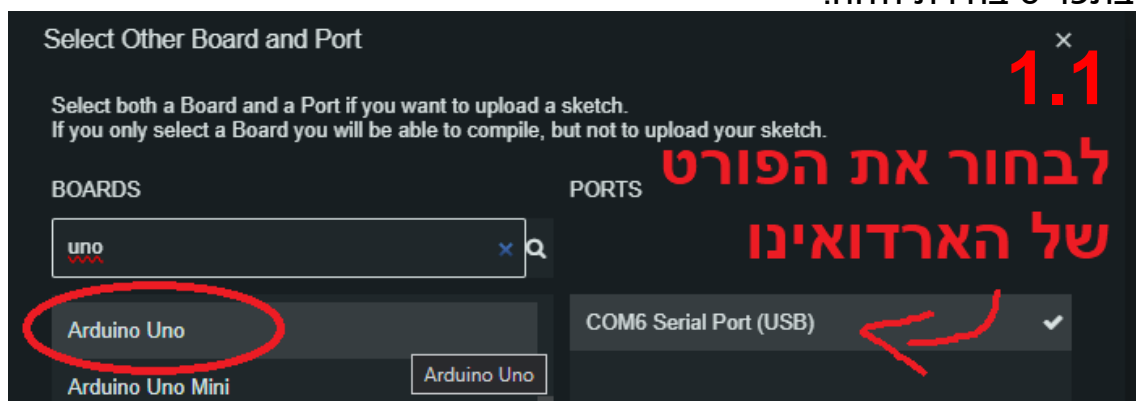
[Arduino Reference.](#)

מעבדה 0

1. הורידו [Arduino IDE](#), והתקינו אותו על המחשב. (ישנה סביבה מקוונת, מוזמנים להתנסות, אך אני לא אתמוך בה...).
2. פתחו את מעבדה 0, בחרו את הלוח (ארדואינו אונו), ובצעו קמפול והעלאה. אם הכל עובד הנורה על הארדואינו צריכה להבהב. (ניתן ללחוץ ישר על העלאה, והוא יבצע גם קמפול לפני).



ובתפריט בחירת הלוח:



3. אם הכל עובד – ניתן להמשיך הלאה.

הרצאה 1

הקדמה על ארדואינו ו-atmega328. הפינים השונים בארדואינו – פלוס, מינוס, והמספור של השאר. לד מובנה. פונקציות setup ו-loop. פונקציות pinMode ו-digitalWrite.

מעבדה 1A

נעביר את הקוד של המעבדה הקודמת לשימוש בלד חיצוני:

1. הפכו את הקוד ממעבדה 0 ליותר גנרי – עשו שימוש ב**משתנים** או ב-**DEFINE**ים.
בנוסף, מומלץ להשתמש ב LED_BUILTIN שהוא DEFINE מובנה עבור הפין בו הLED נמצא בלוח הנבחר. (כדי לראות איך DEFINEים מוגדרים, ניתן לרחף עליהם עם העכבר, או CTRL+קליק ימני, או CTRL+F12).
2. נסו לשים את ההבהוב בתוך לולאה השולטת על זמן העיכוב, כך שנקבל הבהוב בקצב משתנה.
3. נסו להגדיל ולהקטין את זמני העיכוב. מה הכי מהר שאפשר? הכי לאט?
4. קיימת פונקציה delayMicroseconds(), נסו אותה.

מעבדה 1B

נשפר קצת את הקוד של המעבדה הקודמת.

1. העתיקו את תיקיית lab01A, ושנו את שמה ושם הקובץ בתוכה ל-lab01B.
2. הפכו את הקוד שגורם לנורה להבהב לפונקציה. הקוד אמור להיות טיפה יותר קריא.
3. נסו לשים את ההבהוב בתוך לולאה השולטת על זמן העיכוב, כך שנקבל קצב הבהוב משתנה.

מעבדה 1C

- נניח היינו רוצים נורה אחת שמהבהבת בקצב אחד, ונורה אחרת שמהבהבת בקצב אחר, כיצד היינו עושים זאת? זה בעייתי מאוד בדרך שבה השתמשנו (אך ניתן לעשות זאת, מוזמנים לחשוב כיצד).
- כדי לפתור זאת בצורה יותר כללית, ניגש לבעיה בצורה שונה מפונקציית העיכוב: נשמור חותמת זמן של הפעם הבאה בה אנחנו צריכים להבהב את הנורה (כלומר אם היא דלוקה לכבות אותה, ולהפך). אם 'נבדוק מה השעה' ונראה שהגענו לזמן הזה – סימן שהגיע הזמן לשנות את המצב של הנורה. אז גם נעדכן את חותמת הזמן של הפעם הבאה.
1. קראו את שלד הקוד המובא, השלימו את החסר, והריצו אותו. האם הכל עובד?
 2. הוציאו מהערה את הקוד בלולאה, הריצו וקראו. הקוד מיישם את מה שהוסבר מקודם. האם הכל עובד? מה השוני לעומת השיטה שעשינו מקודם?

3. האם ניתן בשיטה דומה ליישם פונקציה ששקולה לפונקציית העיכוב הרגילה? נסו לעשות זאת.
4. יישמו הבהוב של שלושת הנורות במקביל, עם זמני הבהוב שונים (מובא בתור DEFINE).
- נניח לקוד שמהבהב נורה היה לוקח 50 מילישניות לרוץ, האם היינו מקבלים את אותה תבנית של הבהובים? ואם זה היה 5 שניות במקום? האם ניתן לתקן זאת?
- מה היתרונות לעומת פונקציית העיכוב הרגילה?

הרצאה 2

חיבור כפתורים, נגדי פולדאון, פונקציית digitalRead.

מעבדה 2A

נרצה לקבל קלט מהעולם החיצוני בצורת כפתורים שלוחצים עליהם, ולהציג את התוצאה בעזרת הלדים.

1. הגדירו את הפינים אליהם מחוברים הלדים כפלט, ואת הפינים אליהם מחוברים הכפתורים כקלט עם משיכה למעלה.
2. בצעו קריאה של שלושת הכפתורים לתוך משתנים.
3. כתבו לשלושה לדים לבחירתכם את הקריאה מהכפתורים.
4. הריצו את הקוד. האם הכל עובד?

מעבדה 2C

ננסה להבין איך עובדים הפינים בארדואינו – איך מוציאים אליו ערך ואיך קוראים ממנו ערך. המטרה היא להצליח להבין בקווים כללים את תמונה 13-2 בדף הנתונים.

1. קראו את פסקה 13.1, 13.2.1, ו-13.2.2.
2. ניתן למצוא הסברים ודוגמאות נוספות [פה](#).
3. קראו שוב את 13.2.2. האם ניתן לגרום להבהוב הLED ממקודם לקרות אפילו יותר מהר? מה ההבדל?
4. למתקדמים: אם הבנתם הכל עד עכשיו, נסו ליצור את המעגל עם הכפתורים ממקודם, ללא שימוש בפונקציות digitalWrite, pinMode, digitalRead.

מטרת המעבדה הזאת (שהיא יותר תיאורטית) לתת טעימה מהמאחורי הקלעים של ארדואינו, ולחשוף אותכם טיפה לחומרה עצמה. בנוסף, ניתן לראות את היתרונות של היכרות אינטימית עם החומרה – ניתן להגיע לביצועים גבוהים יותר, ולעשות דברים שאי אפשר לעשות בקוד רגיל של ארדואינו.

מעבדה 5A

במעבדה הזאת נתמקד בזמזם הפסיבי – נרצה לגרום לו לנגן שירים שלמים. נרצה לחלק את הקוד לקבצים על מנת לשמור עליו מסודר יותר. נתון לכם בשלד הקוד הקבצים `melodies.c` ו-`melodies.h`. איפה שיש שימוש בפונקציות של `melodies` יש `include` לקובץ `header`, כנלמד בקורס C. בנוסף, בהמשך ניצור קובץ `melodies_utils.ino`, בו נרשום את מרבית הקוד להמשך המעבדה.

התבוננו בקובץ המנגינות. כל מנגינה מורכבת מהמשתנים הבאים:

- **tempo** – המהירות של המנגינה.
- **length** – מספר התווים במנגינה.
- **notes[]** – התווים עצמם.
- **duration[]** – משכי התווים (תו שלם, חצי תו, וכולי).

בשביל לנגן מנגינה, נרצה לעבור על כל התווים, ובעבור כל תו לנגן אותו למשך הזמן המתאים (מושפע מהמהירות של המנגינה).

1. בקובץ הראשי ישנו `include` עבור `melodies.h`, כיוון שנשתמש בדברים המוגדרים בקובץ זה בהמשך הקוד.
2. ישנו גם קובץ חדש, `melodies_utils.ino`, בו נכתוב את המשך הקוד. אין צורך לעשות `include` לקובץ זה כלל. בגלל הסיומת, ארדואינו ידאג לעשות הכל עבורנו.
3. כתבו את הפונקציה `play_note` המקבלת תדר ומשך זמן במילי-שניות, ומנגנת את התדר הזה לזמזם למשך הזמן הנדרש. השתמשו בפונקציות `tone` ו-`noTone` (למידע נוסף על הפונקציות האלו, בדקו ברפרנס של ארדואינו).
4. ניתן לנגן תו רק נגיד 90% מהמשך שלו, ולהשאיר 10% הפסקה. עניין של העדפה אישית.
5. כתבו את הפונקציה `play_melody` המקבלת קצב, מספר תווים, מערך תווים (קבוע – כלומר `const`), ומערך אורכים (קבוע). שימו לב כי מוגדרים הסוגים `note` ו-`note_duration`. עבור כל תו, מצאו את משך הזמן במילישניות (הזכרו בפונקציה שעושה זאת), ואת התדר המתאים בעבור מספר התו (גם כאן ישנה פונקציה שעושה זאת), וקראו לפונקציה `play_note` שייצרתם מקודם כדי לנגן את התו. שימו לב להתייחס לתו מספר אפס, שמשמעו לא לנגן כלום (אך כן להתעכב).
6. בקובץ הראשי, קראו לפונקציה זו עם מנגינה לבחירתכם, או עם כל המנגינות.

מעבדה 12A

במעבדה זאת נעשה כיף חיים. נממש את [המשחק הזה](#).

1. קראו והבינו את שלד הקוד הנתון. עקבו אחרי ההוראות הנתונות בשלד.

אציג איך אני בחרתי לממש אותה, אך הרגישו חופשי לחרוג מכך.
אני חילקתי את הקוד למספר פונקציות עיקריות:

- `init_game` – מאתחל משחק חדש.
 - `wait_button_press` – מחכה ללחיצת כפתור הבאה ומחזיר את מספר הכפתור שנלחץ.
 - `wait_button_release` – מחכה לוודא שכל הכפתורים לא לחוצים למשך זמן ארוך. זאת מכיוון שאני לא רוצים להחשיב לחיצת כפתור אחת ארוכה בתור שלוש לחיצות קצרות, לכן אחרי כל לחיצת כפתור אני מחכה גם לראות שכל הכפתורים כבר לא לחוצים יותר.
 - `show_moves` – מראה לשחקן את הרצף הנוכחי שעליו ללחוץ.
 - `read_moves` – קורא מהשחקן את הניסיון להעתיק את הרצף, ומחזיר את התוצאה (כלומר אם השחקן הצליח או לא).
 - `game_over` – מציג אנימציה של הפסד במשחק ומאתחל משחק חדש.
 - `game_won` – מציג אנימציה של ניצחון במשחק ומאתחל משחק חדש.
- הלוגיקה הראשית של המשחק מתרחשת בלולאה הראשית, ומשתמש בכל הפונקציות שנכתבו לעיל.
2. אם ברצונכם לקבל מספרים אקראיים חדשים כל פעם, ניתן לאתחל את המספרים האקראיים באמצעות קריאה אנלוגית מפין מנותק – שהיא באמת אקראית. עוד מידע ברפרנס.