



מגמת הנדסת אלקטרוניקה ומחשבים

מערכות משובצות מחשב

תוכן עניינים

דבר המפמ"ר.....	2
פרק 1: יסודות התכנות בסביבת מיקרו-בקרים.....	3
פרק 2: כתיבה וקריאה של מפתחים ספרתיים.....	29
פרק 3: מבואות ומוצאים תקביליים.....	60
פרק 4: מיקרו-בקרים.....	85
פרק 5: פסיקות חומרה ומונים.....	101
פרק 6: תקשורת טורית UART בין מיקרו-בקר למחשב.....	116

דבר המפמ"ר

אנו עומדים בפתחה של תקופה של חידושים טכנולוגיים, אתגרים חשיבתיים והתנסויות מרתקות. במגמת הנדסת אלקטרוניקה ומחשבים אתם תחשפו לסביבת למידה מתקדמת ומשמעותית. זכרו, כי אתם חלק חשוב מהמחר ובידכם הכוח והידע לקחת חלק בפיתוח דור העתיד.

הפקדתי בידי צוות מורי המגמה, אחריות רבה, לחשוף אתכם לידע חדשני, להנחות אתכם בפרויקטים מאתגרים ולהוביל אתכם להצלחה.

שלומי אדמונד אחנין

מפמ"ר מגמות הנדסת אלקטרוניקה ומחשבים ומערכות בקרה ואנרגיה

פרק 1: יסודות התכנות בסביבת מיקרו-בקרים

המונח "מחשב" מתייחס למערכת שלמה הבנויה מחלקים שונים. במבט ראשון נראה שהמחשב הטיפוסי, שולחני או נייד, מורכב ממסך (צג), רמקולים, לוח מקשים (מקלדת) ועכבר. למחשב לוח (טאבלט) מרכיבים דומים לאלה של מחשב נייד ומחשב שולחני, פרט לכך שבמקום מקלדת ועכבר משתמשים בו במסך מגע.

שאלה לדיון

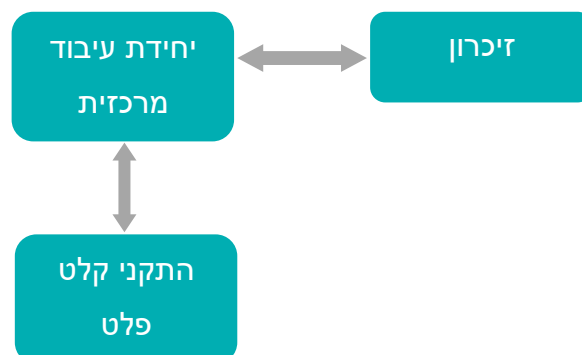
האם סמארטפון (טלפון "חכם") הוא מחשב?

ההסתכלות במחשב כמורכב מצג, מקלדת ועכבר היא מנקודת מבט של המשתמש במחשב. אם נפתח את המארז של מחשב, מה נגלה בתוכו? מה הן היחידות המרכיבות אותו?

1.1 מבנה המחשב

מרכיבי המחשב

בכל מערכת מחשב שלושה חלקים בסיסיים: יחידת עיבוד מרכזית, זיכרון והתקני קלט ופלט.



יחידת עיבוד מרכזית

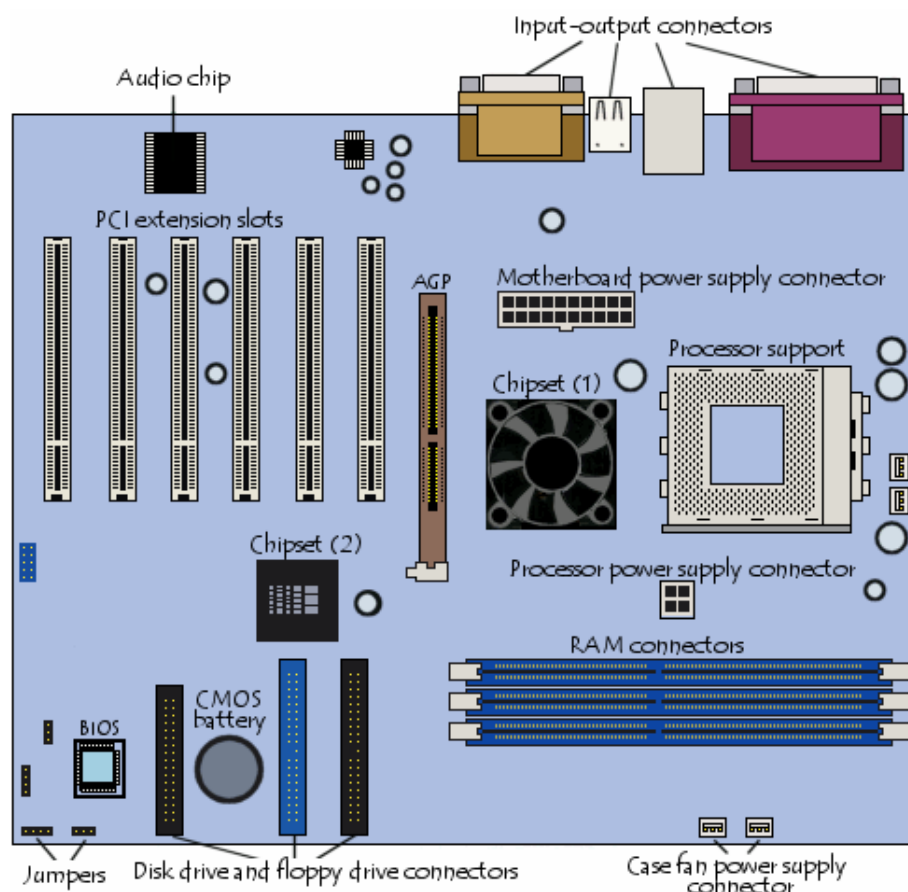
ליחידה זו, הידועה גם בשם מעבד או יע"מ, שני תפקידים חשובים: ביצוע פעולות חישוב ושליטה במרכיבי המערכת האחרים.

זיכרון

יחידת הזיכרון מאחסנת ("שומרת") את התוכניות והנתונים הנחוצים להפעלת המחשב. יחידה זו מבוססת על שבבים אלקטרוניים בלבד.

התקני קלט/פלט

התקני הקלט והפלט משמשים בין היתר ליצירת ממשק (חיבור) בין המשתמש (אדם) למכונה (מחשב). להתקנים אלה מחוברים הצג, המקלדת, העכבר, כרטיס הקול, כרטיס הרשת ועוד. במחשב שולחני או במחשב נייד היחידות האלה מחוברות זו לזו באמצעות לוח אם.



אפשר לחבר ללוח אם מעבדים שונים, לשנות את הרכב הזיכרון ולהחליף את התקני הקלט/פלט.

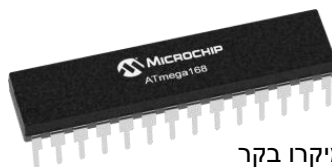
במחשב טיפוסי אפשר להתקין מערכות הפעלה שונות: Windows או גרסאות שונות של Linux. לאחר הפעלת המחשב אפשר להריץ בו תוכניות שונות לפי הצורך.

1.2 מיקרו בקר

כיום נכללת מערכת מחשב ביותר ויותר מכשירים, ובהם מקררים, מכונות כביסה ונתבים. כל אחד מהמכשירים האלה מופעל למטרה מסוימת בלבד, ולכל אחד מהם ממשק משתמש הדרוש להפעלתו ותוכנית ייעודית המשרתת את מטרת המכשיר. אין כל אפשרות להשתמש

באותה מערכת לצרכים אחרים. אין דרך לשנות את הרכב המערכת והיא מופיעה על לוח יחיד. בכל אחד מהמכשירים האלה כלול מחשב זעיר, הנקרא **מיקרו בקר** (Micro Controller Unit). במיקרו-בקר נכללות בשבב יחיד שלוש היחידות המרכיבות כל מערכת מחשב ויחידות משלימות שנציג בפרקים הבאים.

יש מגוון גדול של מיקרו בקרים מארכיטקטורות שונות ולחלקם תכונות מתקדמות: תמיכה ברשתות תקשורת נתונים, תקשורת USB, חיבור התקנים אנלוגיים ועוד. אחת מהארכיטקטורות הנפוצות מוכרת בשם AVR® enhanced RISC. כמה וכמה חברות ייצרו לוח אלקטרוני המבוסס על מיקרו בקר מארכיטקטורה זו.



איור 1.1 מיקרו בקר



איור 1.2 לוח אלקטרוני

1.3 לוחות Arduino

חברת Arduino היא מהמובילות בעולם בתחום קוד פיתוח של חומרה ותוכנה

(Open source hardware and software). החברה מציעה מגוון [לוחות](#) ברמות שונות:

- **רמת כניסה** – משתמש מתחיל;
- **רמה מתקדמת** – משתמש הצריך מהירות גבוהה יותר, מספר הדקים גדול יותר ותכונות מתקדמות;
- **אינטרנט של דברים IOT** – יישומים המשתמשים בתקשורת קוויית או אלחוטית.

בטבלה הבאה מוצגים לוחות נפוצים בשתי רמות ראשונות.

קטגוריית לוח			דגמים
רמת כניסה – משתמש מתחיל			  
			101 LEONARDO UNO
רמה מתקדמת			  
			DUE ZERO MEGA

טבלה 1.1 רשימה חלקית של לוחות

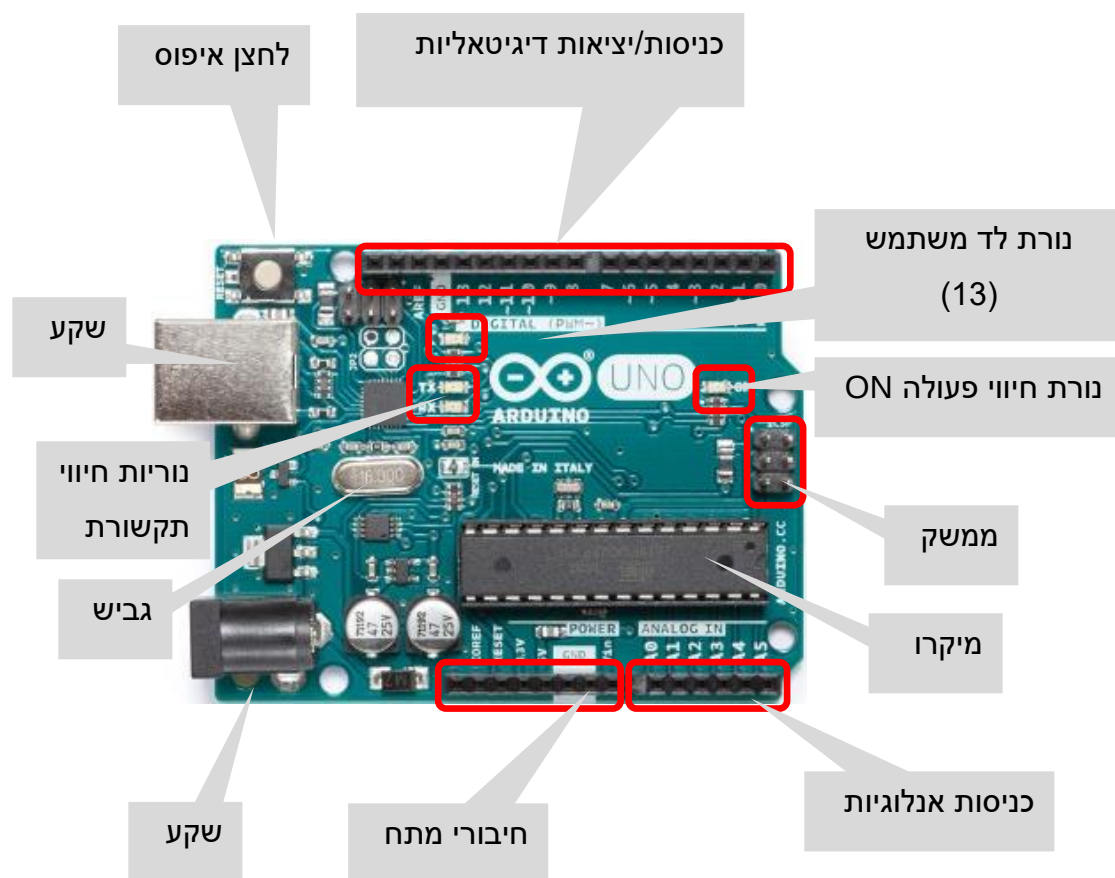
ביחידת המבוא למערכות משובצות מחשב נתמקד בלוח Arduino UNO ונלמד את מאפייניו ואת מאפייני המיקרו-בקר המשובץ בו.

Arduino UNO 1.4

אחד הלוחות הנפוצים של היצרנית Arduino ברמת הכניסה למשתמש המתחיל הוא הדגם **UNO**. ללוח זה משתמשים רבים ויש תיעוד רב ודוגמאות באינטרנט.

לוח UNO מבוסס על המיקרו-בקר ATmega328P. יש לו 14 הדקי כניסה/יציאה דיגיטליים, שש כניסות אנלוגיות, מתנד גבישי בתדר 16MHz, חיבור USB, שקע אספקת מתח, לחצן איפוס וממשק חיבור ICSP.

נוסף על אלה יש בלוח ארבע נורות LED: נורת חיווי פעולה ON, נורה לשימוש כללי המחוברת להדק 13 ושתי נורות לחיווי פעולה של תקשורת בין המיקרו-בקר למחשב.



איור 1.3 תיאור חלקי לוח UNO

מפרט טכני

מיקרו בקר	ATmega328P
מתח פעולה	5V
מתח מבוא בשקע המתח (מומלץ)	7-12V
הדקי קלט/פלט ספרתיים (דיגיטליים) I/O	14 (מהם שישה משמשים כיציאות PWM) ממוספרים מ-0 עד 13
הדקי I/O PWM	שישה מ-14 הדקי I/O ומסומנים ב-~ 3, 5, 6, 9, 10, 11
הדקי I/O תקשורת	TX – 1, RX – 0 (מ-14 הדקי ה-I/O)
כניסות תקביליות (אנלוגיות)	שש מסומנות A0 עד A5
זרם לכל הדק I/O	20mA
זרם להדק 3.3V	50mA
תדירות שעון	16MHz
נורת LED מובנית LED_BUILTIN	הדק 13 בלוח
זיכרון FLASH תוכנית	32KB
זיכרון RAM סטטי לנתונים	2KB

בשלב זה של הלימוד חלק מהמידע במפרט הטכני אינו מוכר לכם. בפרקים הבאים נלמד נושאים הקשורים למערכות ממוחשבות ושם נדון בכל אחד ממרכיבי המפרט שהוצג לעיל:

- בפרק 2 נלמד על מפתחי קלט/פלט ספרתיים
- בפרק 3 נלמד על מפתחי קלט/פלט תקביליים
- בפרק 4 נעמיק במיקרו בקר ATmega328P והארכיטקטורה שלו
- בפרק 5 נלמד על פסיקות חומרה
- בפרק 6 נלמד על תקשורת טורית

1.5 מיקרו בקר ATmega328

כאמור מיקרו בקר הוא מחשב זעיר הכולל בתוכו את שלושת מרכיבי היסוד: מעבד, זיכרון והתקני קלט/פלט. כל מערכת מחשב כוללת בתוכה מרכיבים נוספים ובהם מונים המשמשים קוצבי זמן ומניית אירועים, יחידת תקשורת טורית המאפשרת קישוריות למערכות אחרות, מנגנון המרה של אות מתח אנלוגי לאות ספרתי ומנגנון פסיקה המאפשר להתקני חומרה לפנות למעבד באופן אקראי ולהודיע על אירוע הדורש התייחסות.

המיקרו-בקר ATmega328P של היצרנית Atmel® מבוסס על ארכיטקטורת AVR®. מיקרו בקר זה מיוצר בטכנולוגיית CMOS בעלת צריכת הספק נמוכה ופועל עם מידע ברוחב 8 סיביות (8-bit). ההוראות בו מתבצעות במחזור פעולה יחיד, כלומר עבור קצב של 16MHz יתבצעו כ-16 מיליון פעולות בשנייה.

תכונות

- מעבד 8 סיביות שרוב ההוראות בו מתבצעות במחזור יחיד
- 23 קווי I/O
- זיכרון Flash בגודל 32KB
- זיכרון SRAM בגודל 2KB
- שני מונים/קוצבי זמן של 8 סיביות
- מונה/קוצב זמן של 16 סיביות
- שישה ערוצי PWM
- שישה ערוצי המרה מאנלוגי לדיגיטלי ברוחב 10 סיביות
- תקשורת טורית USART
- פסיקות חומרה חיצוניות ופנימיות

תרגול

שאלה 1.1

באיזה חלק של לוח UNO נמצאים הזיכרונות?

שאלה 1.2

כמה הדקי קלט/פלט ספרתיים יש בלוח UNO?

שאלה 1.3

כמה הדקי קלט/פלט תקביליים יש בלוח UNO?

שאלה 1.4

בספירה של הדקי I/O ספרתיים והדקי I/O תקביליים בלוח UNO מתקבל ערך 20. האם ערך זה תואם את תכונות המיקרו-בקר? אם לא, איך אפשר להסביר את הפער?

שאלה 1.5

בטבלה 1.1 התייחסנו ללוחות ברמה מתקדמת. גלוש לאתר ומצא מידע על הלוח Arduino Mega. על פי האתר, כמה קווי קלט/פלט ספרתיים וכמה קווי קלט ופלט אנלוגיים יש ללוח זה?

שאלה 1.6

בטבלה 1.1 הצגנו רשימה חלקית של לוחות שחברת Arduino מייצרת. מנה כמה שיותר סיבות לכך שהחברה מייצרת מגוון לוחות.

1.6 סביבת הפיתוח Arduino IDE

כחברה התומכת בקוד פיתוח לחומרה ולתוכנה, היא מספקת באתר האינטרנט שלה סביבת פיתוח התואמת את כל הלוחות שהיא מייצרת.

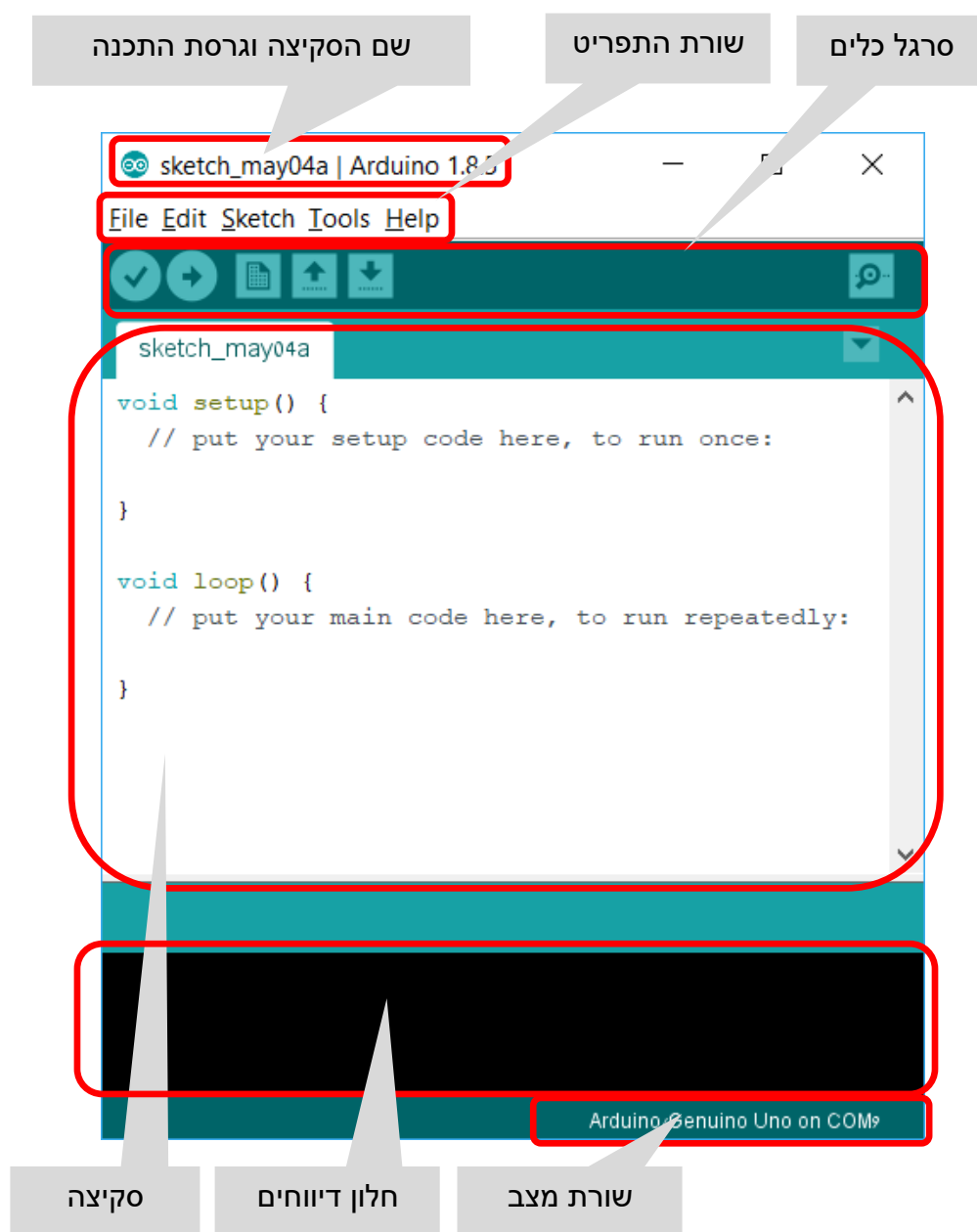
אתרו במחשבכם את סביבת הפיתוח הנקראת Arduino בשולחן העבודה או בתפריטי התוכניות והפעילו אותה.



זיהוי התוכנה לפי הצלמית (icon):

אם סביבת הפיתוח אינה מותקנת, עברו להנחיות התקנה בנספח א.

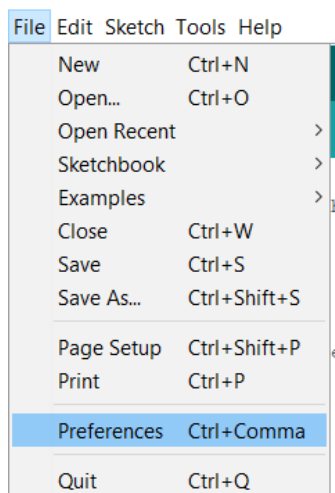
לאחר ההפעלה נקבל חלון של סביבת הפיתוח:



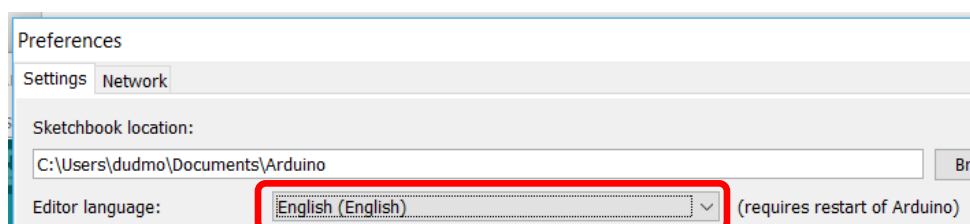
תפריטי סביבת הפיתוח כתובים בשפה האנגלית. מומלץ להישאר בה מכיוון שרוב סביבות הפיתוח הן באנגלית.

שינוי שפת תפריטים

בחרו בתפריט File (קובץ) ולאחר מכן בפריט Preferences (העדפות)



בחלון שיתקבל:



בחרו בשפה המועדפת עליכם ואשרו

לאחר האישור סגרו את הסביבה והפעילו מחדש.

בכל הדוגמאות וההסברים בספר זה נמשיך להשתמש בתפריטים בשפה האנגלית.

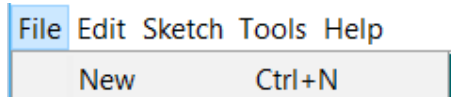
תפריטים וסרגל הכלים

בסרגל הכלים קיצורי דרך לפעולות נפוצות המופיעות בתפריטים השונים.

יצירת סקיצה/תוכנית חדשה



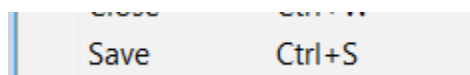
דרך 1: לחיצה על הסמל



דרך 2: תפריט File ולאחר מכן בחירה **New**

בעקבות הפעולה יפתח חלון חדש עם שם המוקצה באופן אוטומטי. בדוגמה שם הסקיצה sketch_may05a.

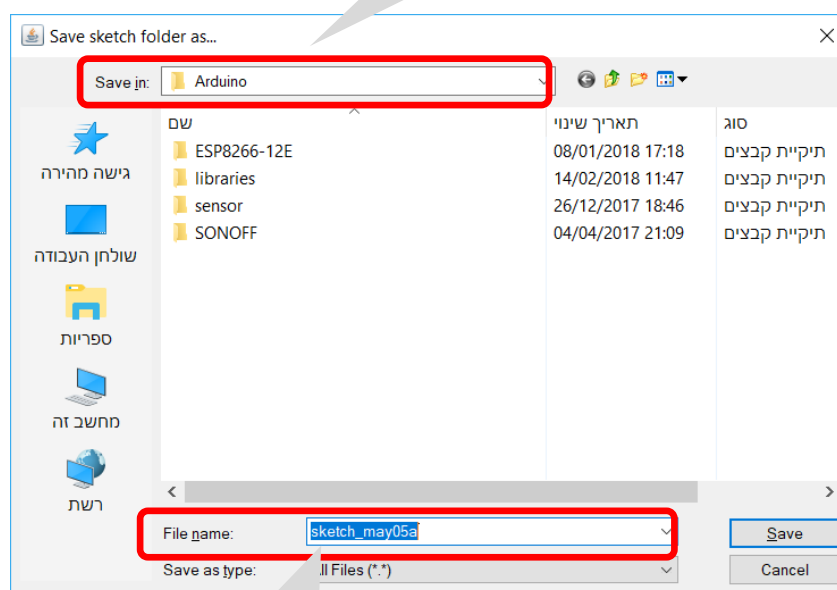
שמירת סקיצה/תוכנית



תפריט File ולאחר מכן בחירה **Save**

אם זו הפעם הראשונה שהתוכנית נשמרת ייפתח חלון ובו תתבקשו לקבוע את המקום שבו היא תישמר ותוכלו לשנות את שמה:

התיקיה בה תשמר התכנית



קביעת שם אחר לתכנית

בחרו תיקייה לאחסון התוכניות שאתם יוצרים וקבעו לה שם מייצג בעל משמעות.

עם שמירת התוכנית בשם שנבחר, סביבת הפיתוח תשמור תיקייה עם אותו שם תוכנית בתיקייה הנבחרת.

1.7 מבנה בסיסי של תוכנית

שפת התכנות ב-Arduino היא C/C++. התחביר של שפה זו הוא כמו התחביר של שפת C, עם תוספות השייכות לתכנות מונחה עצמים בשפת C++. התוכנית עוברת להידור (compile) עבור המיקרו-בקר בשינויים קלים בסקיצה/תוכנית.



במסגרת לימוד שפת התכנות C# הכרנו את מושג **המחלקה** (Class). בכל מחלקה מוגדרות תכונות ופעולות. כידוע, **תכונות** (Fields) הם משתני המחלקה ו**פעולות** (Methods) הן ת-תוכניות המתמחות על פי רוב במטלה יחידה.

בשפת תכנות C++ משתמשים במונחים אחרים: התכונות נקראות **משתני המחלקה** (data members) ופעולות המחלקה נקראות **פונקציות המחלקה** (function members).

המונח **פונקציה** זהה לחלוטין למונח **פעולה**.

בתיעוד באתר Arduino נעשה שימוש במונח פונקציה Functions לתיעוד הפעולות.

בכל יצירה של סקיצה/תוכנית חדשה מופיעה תבנית קבועה:

```
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}
```

בתבנית זו שתי פעולות (פונקציות) בסיסיות: `setup()` ו-`loop()`. פעולות אלה מגיעות ללא תוכן, ועל המשתמש להוסיף את ההוראות המתאימות.

setup() – פעולה הנעשית פעם אחת בלבד, בתחילת ריצת התוכנית, ואחראית לקביעת ערכי ההתחלה למשתנים ולאתחול רכיבי החומרה במיקרו בקר. בפרקים הבאים נלמד על רכיבי חומרה שונים ונלמד כיצד יש לקבוע את אופן פעולתם.

loop() – פעולה שנעשית באופן מחזורי לאחר הפעלת הפונקציה `setup()` בכל פעם מחדש לאחר סיום הפעולה. הפעולה מסתיימת לאחר ביצוע הפונקציה האחרונה בה. פעולת הפונקציה דומה להפעלה בלולאה אין-סופית.

1.8 הוראת פלט הדפסה

כידוע, הוראת הדפסה גורמת להופעת טקסט על המסך. ואולם בלוחות Arduino לא נכללת יחידת תצוגה. כדי לאפשר הצגת טקסט פיתחה החברה אפשרות לשלוח טקסט למחשב שמשתמשים בו בזמן פיתוח התוכנה. שליחת הטקסט נעשית באמצעות כבל ה-USB המקשר את הלוח למחשב.

השימוש בהוראת ההדפסה נועד בין היתר לאפשר למפתח אפשרות לעקוב אחר מהלך ביצוע התוכנית ולאפשר ניפוי שגיאות ביצוע (Debug).

Serial

המחלקה **Serial** מטפלת בכל היבטי ההתחברות לחומרה הקיימת בכל לוח Arduino והשימוש בה. מחלקה זו אחראית לתקשורת עם המחשב דרך כבל ה-USB. שמה של המחלקה נגזר מהאופן שבו המידע מועבר – טורי. בהמשך נדון בעקרונות הפעולה של התקשורת הטורית ונלמד כיצד לנצל אותה למטרות נוספות על הדפסה.

קצב העברת הנתונים

חלק מפעולת היחידה הוא הדרישה לקבוע את מהירות העברת הנתונים בין הלוח למחשב. מהירות העברת הנתונים מיוצגת במספרים. ככל שהמספר גדול יותר קצב העברת הנתונים מהיר יותר. בשלב זה לא נתייחס למספרים מעבר לכך, ובדרך כלל נשתמש בערך 9600. לאחר קביעת הקצב נפתח הערוץ להעברת המידע.

הפעולה הקובעת את קצב העברת הנתונים מתבצעת פעם אחת בלבד ולכן היא מופיעה בפונקציה `setup()`:


```
void setup() {
  Serial.begin(9600);
}
```

קביעת הקצב נקבעת בפעולה **begin**, שהיא חלק מהמחלקה **Serial**.

אופרטור נקודה <.>

כדי להשתמש בפעולה של המחלקה נשתמש בסימן הנקודה <.> כמפריד. סימן זה מקורו בתכנות מונחה עצמים ומיועד להפריד בין שם העצם או המחלקה לבין התכונה או הפעולה שרוצים להפעיל על העצם. לדוגמה, ההוראה `Serial.begin(9600)` מורה למחשב להפעיל את הפעולה `begin` של המחלקה `Serial` ולספק לה את הנתון 9600.

הדפסת טקסט למסך – `print`

במחלקה `Serial` יש הוראה נוספת לשידור מידע למסך המחשב (או לכל התקן אחר) בפעולה **print**. הפעולה מקבלת ערך להדפסה ודואגת לשדר אותו אל היעד.

תחביר הפעולה – אופן השימוש בהוראה:

```
Serial.print(val);
```

`val` – הערך להדפסה

להצגת הודעת טקסט נתחם את הטקסט בגרשיים.

לדוגמה, ההוראה לשדר את הטקסט `Hello, world` היא:

```
Serial.print("Hello, world");
```

דוגמה 1.1: תוכנית להצגת טקסט

```
void setup() {  
  Serial.begin(9600);  
  Serial.print("Setup");  
}  
  
void loop() {  
  Serial.print("Hello, world");  
}
```

הסבר

פעולת setup

התוכנית תקבע באמצעות הפעולה Serial.begin את קצב השידור ל-9600, תפתח את הערוץ לתקשורת ולאחר מכן תשדר את הטקסט Setup. הוראה זו תתבצע פעם אחת בלבד.

פעולת loop

התוכנית תשדר ללא הפסקה את הטקסט Hello, world.

1.9 חיבור לוח UNO והרצת תוכנית

השלב הבא לאחר כתיבת התוכנית בסביבת הפיתוח הוא לבדוק את תקינותה, "להוריד" אותה ללוח ולהריץ אותה.

שלב אימות התוכנית (Verify/Compile) ניתן לביצוע גם כאשר הלוח אינו מחובר.

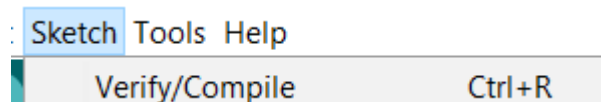
כאשר הלוח מחובר, הוראת "הורד" תבצע את הבדיקה ולאחר מכן את ההורדה.

אימות התוכנית



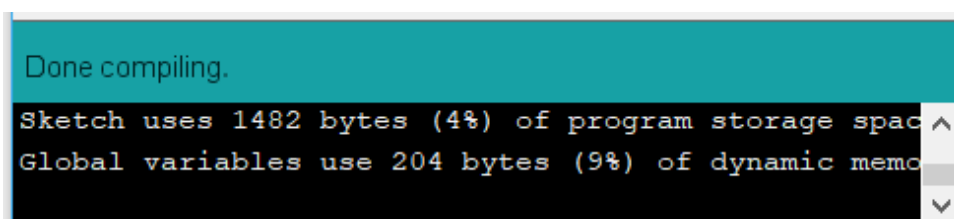
דרך 1: לחיצה על הסמל

דרך 2: תפריט Sketch ולאחר מכן בחירה Verify/Compile



לאחר האימות יופיעו הודעות בחלון הדיווחים – הודעת הצלחה או כישלון בתהליך האימות.

חלון ההודעות כאשר האימות הצליח:



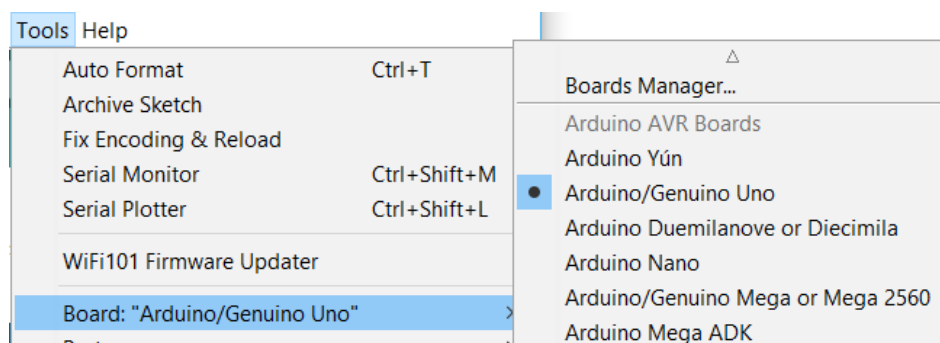
חיבור לוח UNO למחשב

חברו את כבל ה-USB מצידו האחד ללוח ומצידו האחר לשקע פנוי במחשב.

אם החיבור תקין, תידלק נורת חיווי הפעולה (צבע ירוק). אם היא אינה נדלקת נסו שקע אחר במחשב.

בחירת לוח UNO

בתפריט Tools בחרו בפריט Board:



שימו לב שהלוח UNO כבר נבחר.

אם הלוח אינו מסומן, בחרו בו.

אפשר לראות את פרטי הלוח הפעיל בשורת המצב:

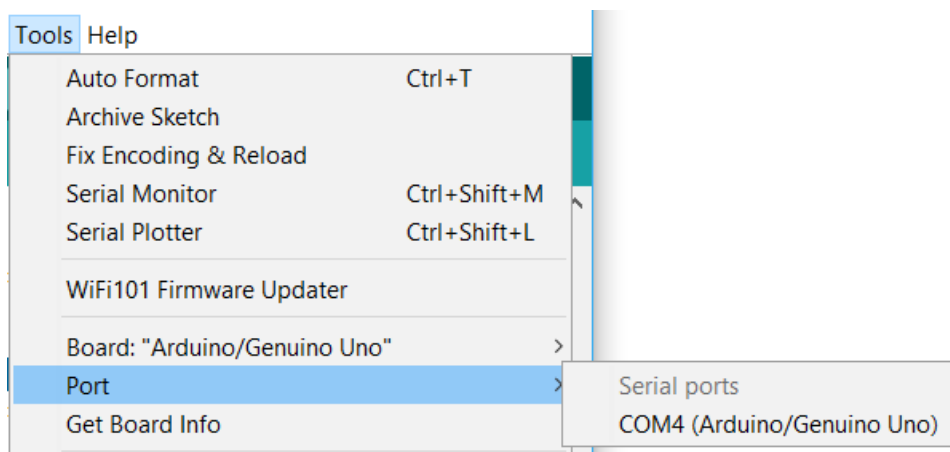


אם הלוח כבר מופיע, אין צורך לקבוע את הלוח.

בחירת מפתח תקשורת

מפתח התקשורת (Port) הוא החיבור בין המחשב ללוח Arduino. מפתח זה מוגדר כערוץ תקשורת במחשב באמצעות כיתוב מתאים com.

בתפריט Tools בחרו בפריט Port:



על פי דוגמה זו נבחר בערוץ תקשורת COM4, המתאים ללוח. ודאו בשורת המצב.

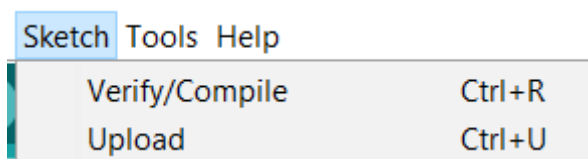
טעינת התוכנית ללוח

טעינה (Upload) היא העברת הפקודות שנערכו בסביבת הפיתוח אל הלוח לשם הרצת התוכנית.



דרך 1: לחיצה על הסמל

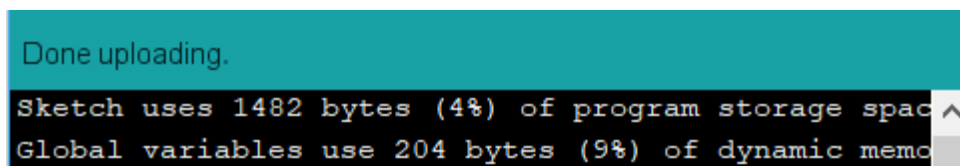
דרך 2: תפריט Sketch ולאחר מכן בחירה Upload



בזמן טעינת התוכנית ללוח יופיע בחלון ההודעות פס התקדמות ואימות ההורדה:



בסוף התהליך תתקבל ההודעה:



לאחר שלב זה התוכנית נמצאת במצב ריצה.

אפשר להריץ אותה מחדש באמצעות לחצן האיפוס שבלוח.

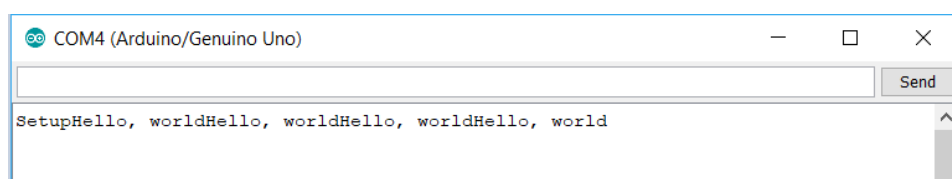
חלון המסוף Serial Monitor

כפי שלמדנו, הפעולה Serial.print משדרת טקסט למחשב. יש תוכנות שיודעות לקרוא מידע שנשלח למחשב. בסביבת הפיתוח נכלל מרכיב המתחבר באופן אוטומטי לערוץ התקשורת במחשב המציג את המידע המגיע מהלוח. מרכיב זה נקרא **Serial Monitor**.

דרך 1: לחיצה על הסמל  הנמצא מצד ימין של סרגל הכלים

דרך 2: תפריט Tools ולאחר מכן בחירה Serial Monitor

בכל פעם שנפעיל את המסוף תתאפס התוכנית ותתחיל מההתחלה.



שימו לב שהמילה Setup מופיעה פעם אחת בלבד, מכיוון שההוראה להצגתה מופיעה בפעולה setup, ואילו הטקסט Hello, world מופיע כמה פעמים. תמונת המסוף מציגה את תוצאת ההרצה לזמן קצר בלבד. הטקסט ימשיך להופיע ללא הפסקה.

דוגמה 1.2: תוכנית להצגת טקסט בשורות שונות עם השהיה

```
void setup() {  
  Serial.begin(9600);  
  Serial.print("Setup\n");  
}  
  
void loop() {  
  delay(1000);  
  Serial.print("Hello, world\n");  
}
```

הסבר

הוספנו לתוכנית שני פריטים:

תו סיום שורה חל: בכל מקום בטקסט שמשולב בו צמד סימנים זה, כל מה שאחריו יוצג בשורה חדשה.

השהיה delay: פעולה המעכבת את המשך ביצוע התוכנית על פי ערך מספרי. ערך מספרי 1000 מציין עיכוב של שנייה אחת. עיכוב מינימלי הוא של אלפית השנייה.

פעולת setup

התוכנית תקבע בהוראה Serial.begin את קצב השידור ל-9600, תפתח את הערוץ לתקשורת ולאחר מכן תשדר את הטקסט Setup בשורה אחת.

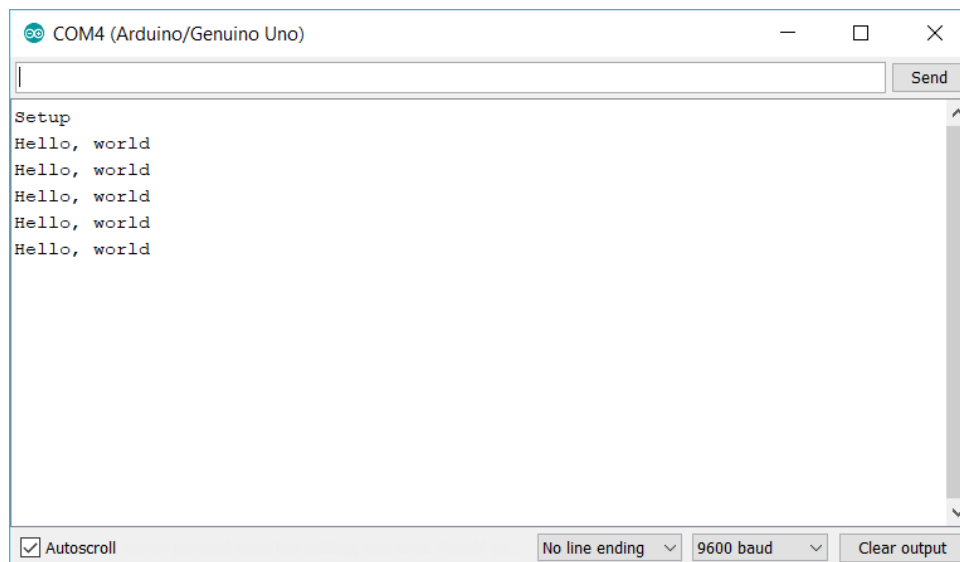
הוראה זו תתבצע פעם אחת בלבד.

פעולת loop

עיכוב התוכנית לשנייה אחת.

שידור הטקסט Hello, world ומעבר לשורה חדשה.

תוצאת ריצת התוכנית כפי שנצפתה במסוף:



דוגמה 1.3: תוכנית להצגת טקסט בשורות עשר פעמים

```
int counter;          // משתנה מונה
void setup() {        // פעולה אתחול
  Serial.begin(9600);  // פתיחת ערוץ תקשורת
  counter = 0;         // איפוס המונה
}
void loop() {         // פעולה הלולאה
  if (counter < 10) {  // אם ערכו של המונה קטן מ-10
    Serial.print("Hello\n"); // שדר טקסט
    counter++;         // קדם את המונה ב-1
  }
}
```

הסבר

הגדרת משתנים

counter משתנה שלם המשמש מונה. המשתנה מוגדר לפני הפעולות ולכן אפשר להשתמש בו בשתיהן.

פעולה setup

התוכנית תקבע באמצעות ההוראה Serial.begin את קצב השידור ל-9600 ותפתח את הערוץ לתקשורת.

איפוס המונה (counter).

פעולה loop

פעולה שהיא בעצם לולאה אין-סופית. בכל הפעלה שלה נבדק ערכו של המונה. אם ערכו קטן מ-10 תוצג הודעת טקסט וערכו של המונה יקודם ב-1.

כעבור 10 הפעלות של הפעולה ערכו של המונה יהיה 10, ואז התנאי יפסיק להתקיים. מרגע שהתנאי חדל להתקיים לא יוצג טקסט במסוף.

דוגמה 1.4: תוכנית להצגת מספרים מ-10 עד 20

```
int counter;
void setup() {
  Serial.begin(9600);
  counter = 10;
}
void loop() {
  if (counter <= 20) {
    Serial.print(counter); // הצגת ערך משתנה
    Serial.print("\n");
    counter++;
  }
}
```

הסבר

הגדרת משתנים

counter משתנה שלם המשמש מונה. המשתנה מוגדר לפני הפונקציות ולכן אפשר להשתמש בו בשתיהן.

פעולה setup

התוכנית תקבע באמצעות ההוראה Serial.begin את קצב השידור ל-9600 ותפתח את הערוץ לתקשורת.

אתחול המונה ל-10.

פעולה loop

פעולה זו היא בעצם לולאה אין-סופית. בכל הפעלה שלה נבדק ערכו של המונה. אם ערכו קטן או שווה ל-20 יוצג ערכו של המונה, וערכו יקודם ב-1.

כעבור 11 הפעלות של הפעולה ערכו של המונה יהיה 21, ואז יפסיק התנאי להתקיים. מרגע שהתנאי חדל להתקיים לא יוצג טקסט במסוף.

שימו לב! השתמשנו בהוראה print גם כדי להציג ערך משתנה.



שאלה 1.5

כתבו תוכנית המציגה את הפרטים הבאים פעם אחת בלבד. יש להציג את הפרטים בשורות שונות.

שם פרטי ושם משפחה

כיתה

בית ספר

פתרו בשתי דרכים שונות: באמצעות הפעולה setup ובאמצעות משתנה מונה.

שאלה 1.6

כתבו תוכנית המציגה 20 פעמים את הפרטים משאלה 1.5.

שאלה 1.7

כתבו תוכנית המציגה את שמו של אחד מכם עשר פעמים ולאחר מכן את שם בית ספרכם.

שאלה 1.8

כתבו תוכנית המציגה את המספרים מ-10 עד 100 בשורות שונות.

שאלה 1.9

כתבו תוכנית המציגה שעון שניות המתחיל ב-10 ומסתיים ב-99. כל הצגה תהיה בשורה נפרדת ובין הצגה להצגה תהיה השהיה של שנייה אחת.

שאלה 1.10

נתונה התוכנית:

```
int counter;
void setup() {
  Serial.begin(9600);
  counter = 1;
}
void loop() {
  if (counter <= 40) {
    if (counter % 6 == 0) {
      Serial.print(counter);
      Serial.print("\n");
    }
    counter++;
  }
}
```

עקבו אחר התוכנית ורשמו את הפלט המתקבל לאחר הרצתה.

שאלה 1.11

נתונה התוכנית:

```
int counter;
void setup() {
  Serial.begin(9600);
  counter = 1;
}
void loop() {
  if (counter <= 15) {
    Serial.print(counter);
    if (counter % 5 == 0) {
      Serial.print("\n");
    }
    else {
      Serial.print(", ");
    }
    counter++;
  }
}
```

עקבו אחר התוכנית ורשמו את הפלט המתקבל לאחר הרצתה.

סיכום פרק 1

בפרק 1 למדתם את הנושאים האלה:

1. מיקרו בקר וההבדל בינו ובין מחשב
2. לוח משובץ מחשב
3. קוד פיתוח חומרה ותוכנה
4. מאפייני לוח Arduino UNO
5. מאפייניו העיקריים של המיקרו-בקר ATmega328P
6. סביבת הפיתוח של Arduino
7. מבנה תוכנית בסביבת Arduino
8. מחזור פיתוח תוכנית: שמירה, אימות וטעינה של תוכנית
9. הוראות להצגת מידע בערוץ תקשורת טורית
10. שימוש במסוף Serial Monitor להצגת מידע
11. שימוש במשתנים ובמשפט תנאי

פרק 2: כתיבה וקריאה של מפתחים ספרתיים

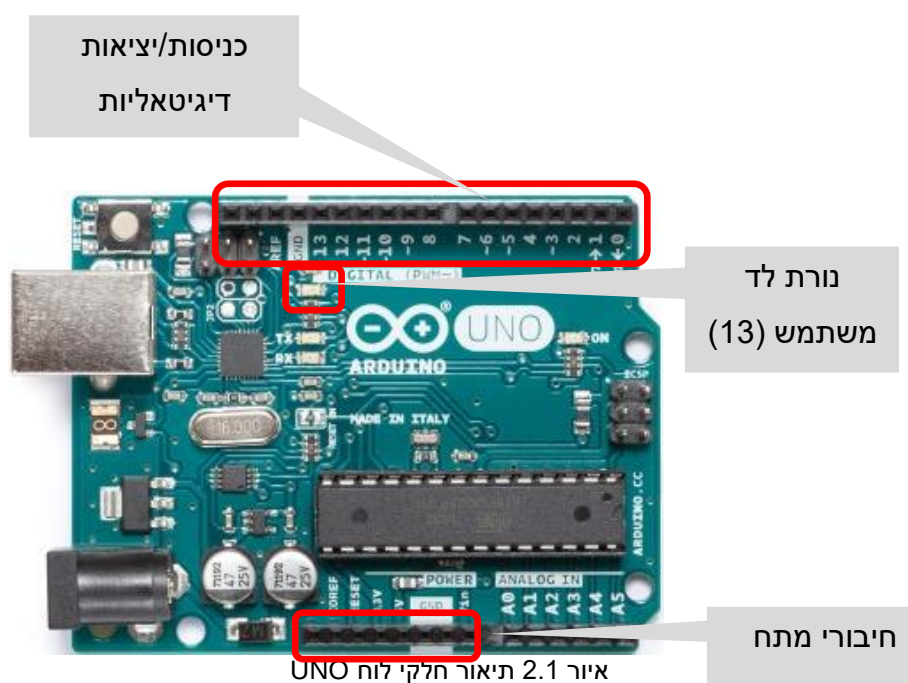
ביחידה הקודמת הכרנו את סביבת הפתוח של Arduino והרצנו בה תוכנית עם שתי פונקציות:

- `setup()`, המופעלת פעם אחת בתחילת ריצת התוכנית.
- `loop()`, המופעלת בצורה מחזורית (בעצם, כלולאה אינסופית).

בתוך הפונקציות השתמשנו ברכיב תוכנה Serial ששימש אותנו להדפסת טקסט (מחרוזות) למסוף (Serial monitor) של סביבת הפיתוח באמצעות ההוראה `Serial.print`.

הכרנו את מבנה לוח Arduino UNO שנכללות בו כמה יחידות.

ביחידה זו נתמקד ביחידות המסומנות באיור:



2.1 מפתח ספרתי

ביחידה הקודמת למדנו שבמערכת ממוחשבת שלושה מרכיבים יסודיים: מעבד, זיכרון והתקני קלט/פלט. תפקיד התקני הקלט/פלט הוא לחבר את ה"עולם" אל המעבד, כלומר להעביר מידע ממקורות חיצוניים למעבד ולהפך.

דוגמה – הפעלה אוטומטית של תאורה

נניח כי מעוניינים לחסוך בהוצאות באנרגיה חשמלית ולהפעיל תאורה באזור מסוים רק כאשר היא נדרשת. הפעלת התאורה תיעשה בכל פעם שנזהה תנועה באזור והתאורה תופעל למשך זמן קצוב של 30 שניות.

מה דרוש לשם כך?

1. רכיב שיזהה את התנועה – חיישן;
2. מערכת הפעלה לתאורה – נורת LED;
3. מערכת שתקלוט את המידע ממרכיב זיהוי התנועה ותפעיל את התאורה. לאחר 30 שניות המערכת תכבה את התאורה – בקר.



הבקר מקבל מידע מהחיישן ומורה על הפעלת התאורה למשך הזמן הדרוש.

אם נתייחס לבקר כאל מרכז המערכת, נזהה שהמידע מהחיישן **נקלט** על ידי הבקר, ולאחר מכן מתבצע תהליך **פלט** אל הנורה.

החיישן מזהה אחד משני מצבים: תנועה או חוסר תנועה.

הנורה יכולה להיות באחד משני מצבים: דולקת או כבויה.

בשני המקרים החיישן והנורה יכולים להיות באחד משני מצבים, שנוכל לסמן אותם בערכים 0 או 1. אלה ערכים מספריים ולכן נוכל להתייחס אליהם כאל **אותות ספרתיים (דיגיטליים)**.

החיישן הוא **כניסה (INPUT)** ספרתית אל הבקר, והמפעיל הוא **יציאה (OUTPUT)** ספרתית מהבקר.

כניסה ספרתית – Digital input

הדק (Pin) בבקר שנועד לקלוט מידע מרכיב קלט בעל שני מצבים, 0 או 1.

יציאה ספרתית – Digital output

הדק (Pin) בבקר שנועד לפלוט מידע אל רכיב פלט בעל שני מצבים, 0 או 1.

מפתח ספרתי – Digital port

מפתח ספרתי הוא אוסף של כניסות ויציאות ספרתיות המוגדרות כקבוצה אחת. גודלו של המפתח מוגדר על ידי הבקר. אפשר להגדיר את כל המפתח כקלט, כפלט או כשילוב כלשהו שלהם, לפי הצורך.

2.2 מפתחים ספרתיים בלוח UNO

אם נתבונן מקרוב באיור 2.1 נוכל לראות בלוח את **מחבר כניסות/יציאות דיגיטליות**.

איור 2.2 מראה את המחבר מקרוב. במחבר מופיעים ערכים מספריים בין 0 ל-13. כל אחת מנקודות החיבור יכולה להיות כניסה או יציאה ספרתית.



איור 2.2 תיאור מחבר מפתחים ספרתיים

הגדרת כיוון ההדקים; כניסה או יציאה נעשית באמצעות הוראה מתאימה בשפת C. אפשר להגדיר כיוון לכל הדק בנפרד על פי מספרו הסידורי מ-0 עד 13 או להגדיר את הכיוון של קבוצת הדקים המוגדרת כמפתח אחד. לכל מפתח ניתן שם ייחודי הכולל את קבוצת ההדקים: PB, PC, PD.

למשל PD מתאר את מפתח D – PORT D.

בטבלה שלהלן מתוארים המפתחים השייכים למיקרו-בקר של לוח UNO.

מפתח/סיבית	0	1	2	3	4	5	6	7
PB	8	9	10	11	12	13	C	C
PC	A0	A1	A2	A3	A4	A5	R	
PD	0	1	2	3	4	5	6	7

טבלה 2.1 מיפוי הדקי הלוח למפתחי הבקר

C – הדקי גביש Crystal, R – הדק איפוס Reset

לפי הטבלה, ההדקים בלוח מ-0 עד 7 שייכים למפתח PD, ואילו ההדקים 8 עד 13 שייכים למפתח PB. אפשר להשתמש במפתח PC כמפתח ספרתי או כמפתח אנלוגי, שעליו נרחיב ביחידה הבאה.

שימו לב!

הדקים 0 ו-1 משמשים לתקשורת טורית. קווים אלה משמשים גם את רכיב התוכנה Serial, המשמש להצגת מידע ב-Serial Monitor. שימוש בהדקים אלה למטרות אחרות יפריע לתקשורת זו. נשתדל להימנע משימוש בקווים אלה. אם נשתמש בה ייתכן שלא נצליח להעביר (UPLOAD) את התוכנית למיקרו-בקר וגם לא להשתמש ברכיב Serial.

2.3 מאפיינים חשמליים

כל אחד מההדקים 0 עד 13 יכול להיות קלט או פלט. תכונותיו החשמליות של כל הדק תלויות בכיוון שהוגדר עבורו – כניסה או יציאה.

מאפייני הדק כניסה

סמל	תיאור	ערך מזערי	ערך מרבי	יחידה
V_{IL}	מתח מבוא לרמה נמוכה	-0.5	1.5	V
V_{IH}	מתח מבוא לרמה גבוהה	3	5.5	V
I_{IL}	זרם זליגה במבוא לרמה נמוכה		1	μA
I_{IH}	זרם זליגה במבוא לרמה גבוהה		1	μA

טבלה 2.2: מאפיינים חשמליים, הדק כניסה

עבור $V_{CC} = 5V$

מאפייני הדק מוצא

סמל	תיאור	ערך מזערי	ערך טיפוסי	ערך מרבי	יחידה
V_{OL}	מתח מוצא ברמה נמוכה			0.9	V
V_{OH}	מתח מוצא ברמה גבוהה	4.1			V
I_{OL}	זרם מוצא ברמה נמוכה		20	40	mA
I_{OH}	זרם מוצא ברמה גבוהה		-20	-40	mA

טבלה 2.3: מאפיינים חשמליים, הדק מוצא

עבור $V_{CC} = 5V$

אף שכל הדק יכול להעביר זרם של 20mA, מותר לצרוך זרם כולל שלא יעלה על ערך מסוים, לפי קבוצת הדקים ולפי הרמה הלוגית במוצא:

IOH

קבוצה 1: PC0-PC5, PD0-PD4, ADC7, RESET לכל היותר 100mA

קבוצה 2: PB0-PB5, PD5-PD7, ADC6, XTAL1, XTAL2 לכל היותר 100mA

IOL

קבוצה 1: PC0-PC5, ADC7, ADC6 לכל היותר 100mA

קבוצה 2: PB0-PB5, PD5-PD7, XTAL1, XTAL2 לכל היותר 100mA

קבוצה 3: PD0-PD4, RESET לכל היותר 100mA

2.4 מפתחי פלט

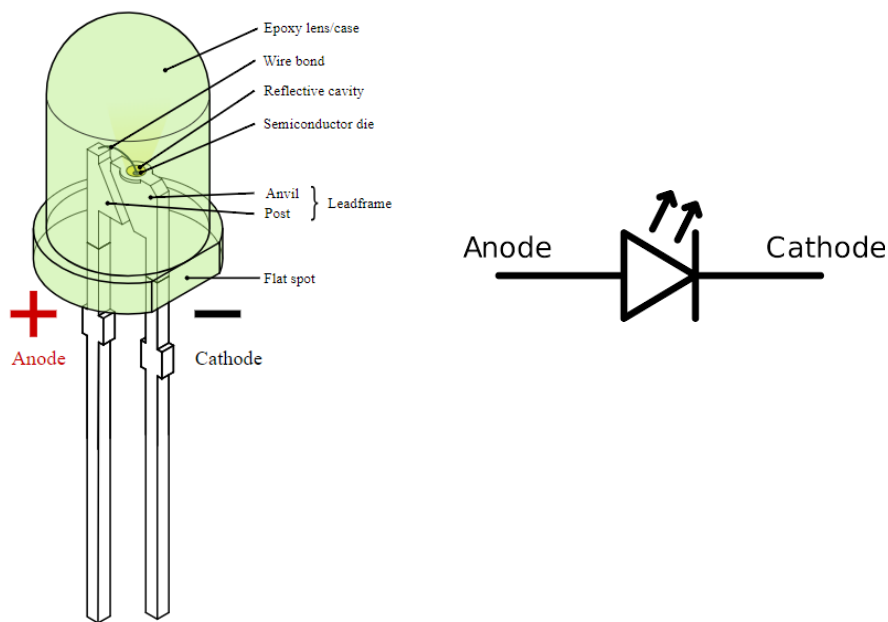
אל מפתח הפלט נחבר רכיבים אלקטרוניים ספרתיים. רכיב ספרתי מופעל בשתי רמות לוגיות בלבד, 0 או 1, 0V או 5V בהתאמה (בפועל רמות המתח יכולות להיות שונות, ראו טבלה 2.3). בסעיף זה נציג אופנים שונים של חיבור נורת LED ותצוגת שבעה מקטעים.

דיודת LED

דיודה פולטת אור Light Emitting Diode היא דיודה שבזמן מעבר זרם דרכה היא פולטת אור בספקטרום צר, כלומר בצבע יחיד. הדיודה משמשת כנורת חיווי, כנורת תאורה, בצגי טלוויזיה ומחשב, בשלט רחוק ובמכשירים רבים נוספים.

מאפייני הדיודה

הדיודה פולטת אור כאשר היא בממתח קדמי ואינה פולטת אור בממתח אחורי או מתחת למתח הפעולה שלה בממתח קדמי. כיוון הזרם הקדמי בדיודה הוא מהאנודה אל הקטודה. עוצמת התאורה של הדיודה תלויה בעוצמת הזרם דרכה. ככל שעוצמת הזרם גבוהה יותר כך גדלה עוצמת התאורה, ולהפך.



איור 2.3 סמל הדיודה LED ומבנה האריזה

מאפייני דיודה טיפוסית RED LED 5mm

ערכי זרם מרביים

Item	Symbol	Value	Unit	Description
Forward Current	I_F	20	mA	זרם הולכה קדימה
Peak Forward Current	I_{FP}	30	mA	זרם שיא בהולכה קדימה
Suggestion Using Current	I_{SU}	16-18	mA	זרם מוצע לשימוש
Reverse Current ($V_R=5V$)	I_R	10	μA	זרם אחורי (זליגה)
Power Dissipation	P_D	105	mW	פיזור הספק

הסבר

- זרם הולכה קדימה מרבי הוא הזרם בעוצמה הגבוהה ביותר שאפשר להעביר באופן רציף בלד ללא גרימת נזק.
- זרם שיא בהולכה קדימה הוא הזרם הרגעי שאפשר להעביר בלד ללא גרימת נזק.
- זרם מוצע לשימוש הוא הערך המומלץ. בפועל נקבע את הערך כך שיתאים לעוצמת ההארה הרצויה.
- הזרם האחורי הוא הזרם המרבי שיזרום בכיוון ההפוך.
- פיזור הספק מרבי הוא הערך הגבוה ביותר שאפשר לצרוך בלד ללא גרימת נזק.

מתח ואורך גל

Item	Symbol	Min	Max	Unit	Description
Forward Voltage	V_F	1.8	2.2	V	מתח ההולכה קדמי
Wavelength	λ	620	625	nm	אורך הגל של האור האדום

הסבר

- מתח ההולכה הקדמי נע בין $1.8V$ ל- $2.2V$, בתלות בעוצמת הזרם העובר בדיודה.
- אורך הגל קובע את צבעו של האור הנפלט.

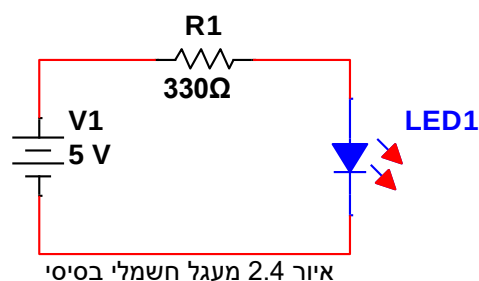
תרגול

שאלה 2.1

חפש באמצעות מנוע חיפש באינטרנט מפרטי דיודת לד RED LED 5mm והשווה עם ערכי דיודה טיפוסית.

מעגל חשמלי בסיסי

באיור שלהלן מעגל טורי הכולל דיודת לד, נגד ומקור מתח. הזרם במעגל נקבע על ידי הנגד ומקור המתח:



דוגמה 2.1 - חישוב הזרם במעגל

נתוני דיודת הלד: $V_F=1.8V$, $I_F=10mA$

על פי חוק המתח: $V_1 = V_R + V_D$

מכאן שמתח הנגד הוא: $V_R = V_1 - V_D = 5 - 1.8 = 3.2V$

מכיוון שהמעגל טורי, זרם הנגד שווה לזרם הדיודה. מתוך חוק אוהם נוכל לחשב את עוצמת הזרם: $I = V_R / R_1 = 3.2 / 330 = 9.7mA$. עוצמת הזרם נמוכה מעט מעוצמת זרם ההולכה הקדמי הנקוב של הדיודה.

שאלה 2.2

חשבו את המתח המתפתח על הנגד במעגל המוצג באיור 2.4 כדי שזרם הדיודה יהיה 3.2mA . נתון: $V_F=1.8\text{V}$.

שאלה 2.3

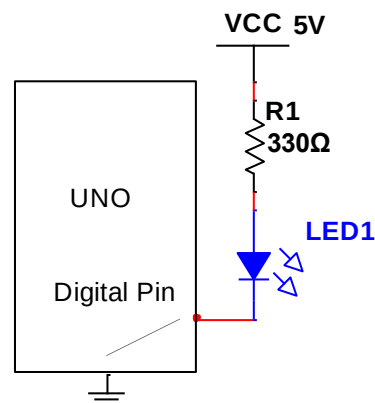
חשבו את המתח המתפתח על הנגד במעגל המוצג באיור 2.4 כדי שזרם בדיודה יהיה 5mA . נתון: $V_F=1.8\text{V}$ ומתח מקור $V_1=3.3\text{V}$.

חיבור דיודת לד למיקרו-בקר

אפשר לחבר למיקרו-בקר דיודת לד ונגד בשתי צורות, כמתואר בשני האיורים שלהלן.

בשני המקרים המעגל החשמלי נסגר דרך מתח האספקה של המיקרו-בקר.

חיבור Pull-Up



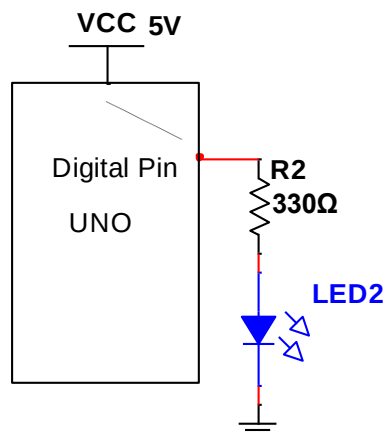
איור 2.5 חיבור Pull-Up למיקרו-בקר

הדק מוצא משמש כהדק השלילי של מקור המתח במעגל. המעגל החשמלי נסגר דרך הדק המיקרו-בקר ומשם עובר הזרם אל הדק האדמה של המיקרו-בקר.

דיודת הלד תדלוק כאשר מוצא ההדק יהיה נמוך ($0\text{V} - \text{LOW}$), מכיוון שיהיה הפרש פוטנציאלים בין מקור המתח V_{CC} להדק.

דיודת הלד תהיה כבויה כאשר מוצא ההדק יהיה גבוה ($5\text{V} - \text{HIGH}$), מכיוון שלא יהיה הפרש פוטנציאלים בין מקור המתח להדק.

חיבור Pull-Down



איור 2.5 חיבור Pull-Down למיקרו-בקר

הדק מוצא המשמש כהדק החיובי של מקור המתח במעגל (אדמה - Ground). המעגל החשמלי נסגר ממקור המתח של המיקרו-בקר דרך הדק המוצא שלו.

דיודת הLED תדלוק כאשר מוצא ההדק יהיה גבוה ($5V - HIGH$), מכיוון שיהיה הפרש פוטנציאלים בין ההדק לאדמה.

דיודת הLED תהיה כבויה כאשר מוצא ההדק יהיה נמוך ($0V - LOW$), מכיוון שלא יהיה הפרש פוטנציאלים בין ההדק לאדמה.

הגדרת הדק מוצא במיקרו-בקר

לפני שימוש בהדק ספרתי יש להגדיר את אופן פעולתו: כניסה או יציאה. הגדרת ההדק נעשית בהוראה pinMode:

הגדרה

`pinMode(pin, mode)`

pin מספר ההדק הספרתי מ-0 עד 13

mode הגדרת אופן הפעולה של ההדק:

OUTPUT - מוצא

INPUT - מבוא

כתיבה להדק מוצא ספרתי

הכתיבה להדק מוצא ספרתי נעשית בהוראה digitalWrite:



digitalWrite(pin, value)

pin מספר ההדק הספרתי מ-0 עד 13

value הערך המסופק להדק:

LOW - ערך נמוך - מתח V0

HIGH - ערך גבוה - מתח V5

* בלוחות שבהם מתח האספקה הוא V3.3 הערך הגבוה הוא V3.3.

דוגמה 2.2 – הפעלת לד בהדק 13

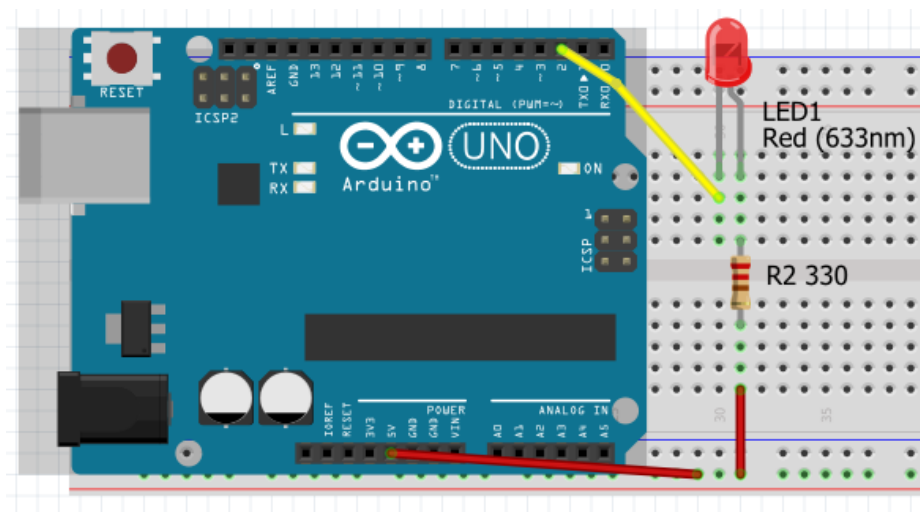
במרבית לוחות Arduino מחוברת להדק 13 דיודת לד. הדיודה מחוברת בתצורת Pull-Down. נגדיר את הדק 13 כהדק מוצא. הדלקת הדיודה תתבצע ברמה גבוהה וכיבוי ברמה נמוכה. בתוכנית הבאה דיודת הלד תהבהב: שנייה במצב דולק ושנייה במצב כבוי.

```
int counter;
void setup() {
  pinMode(13, OUTPUT); // הגדרת הדק 13 כמוצא
}

void loop() {
  digitalWrite(13, HIGH); // הדלקת דיודת הלד
  delay(1000);           // השהיה של 1 שנייה
  digitalWrite(13, LOW);  // כיבוי דיודת הלד
  delay(1000);           // השהיה של 1 שנייה
}
```

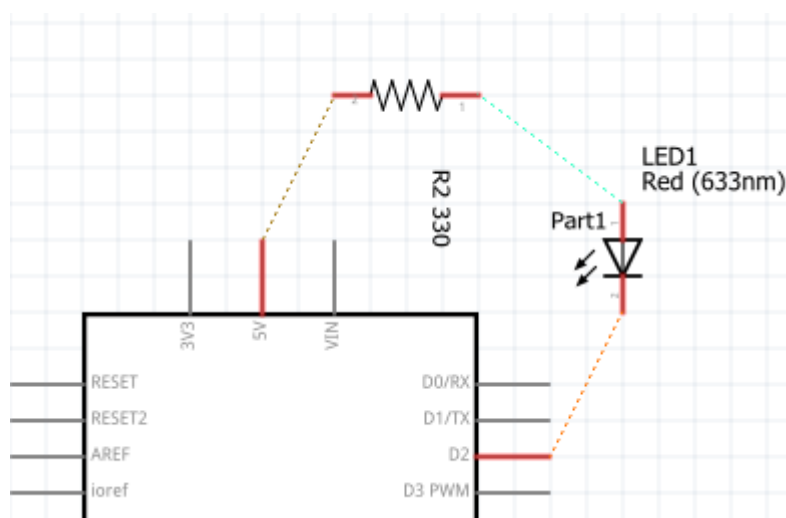
דוגמה 2.3 – חיבור לד להדק 2 בתצורת Pull-Up

באיור שלהלן מתואר חיבור דיודת לד ללוח UNO.



איור 2.6 סכמת חיבורים למיקרו-בקר

להלן דיאגרמה חלקית של המעגל החשמלי:



איור 2.6 תרשים חשמלי למיקרו-בקר

תוכנית להפעלת דיודת הלד – הדלקה למשך שנייה וכיבוי למשך שנייה. שימו לב, מכיוון שהתצורה הפוכה לזו של דוגמה 2, ההפעלה והכיבוי הם ברמות לוגיות הפוכות.

```
void setup() {
    pinMode(2, OUTPUT); // הגדרת הדק 2 כמוצא
}

void loop() {
    digitalWrite(2, HIGH); // כיבוי דיודת הלד
    delay(1000); // השהיה של 1 שנייה
    digitalWrite(2, LOW); // הדלקת דיודת הלד
    delay(1000); // השהיה של 1 שנייה
}
```

תרגול 

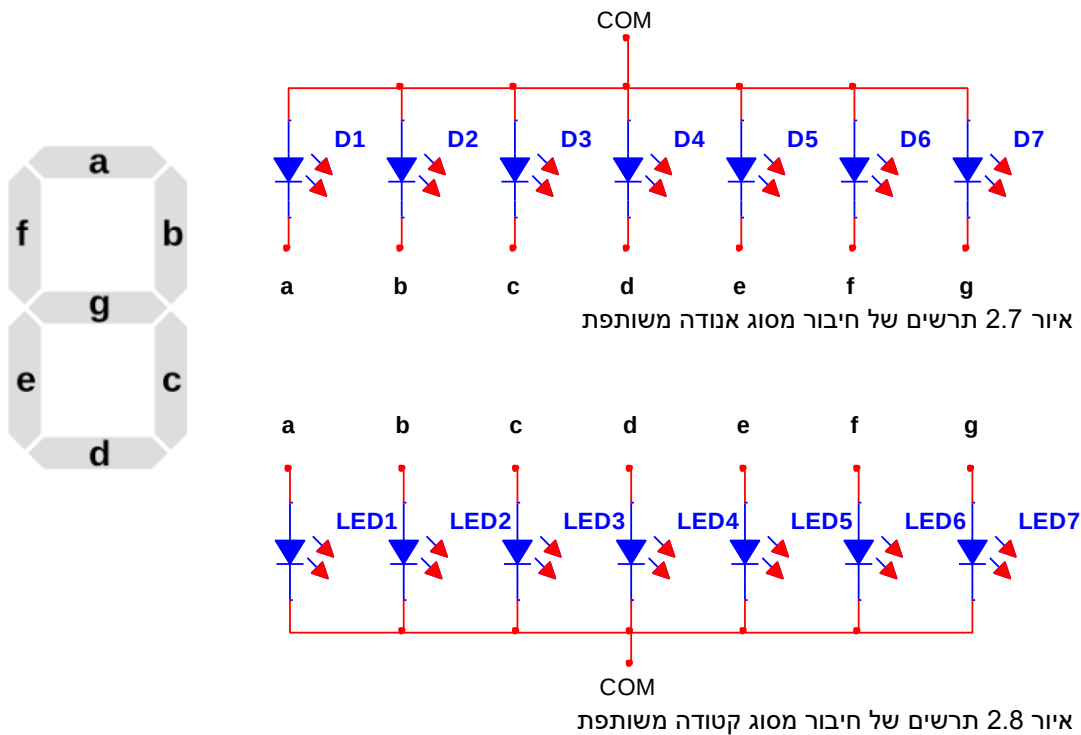
שאלה 2.4

מחברים 2 דיודות לד עם נגדים מתאימים כמו בדוגמאות 1 ו-2 ליציאות ספרתיות 4 ו-5. שתי הדיודות מחוברות בתצורת Pull-Down.

- א. כתבו תוכנית להפעלת שתי הדיודות בו בזמן כך שכאחת דולקת השנייה כבויה ולהפך. כל דיודה תדלוק חצי שנייה ותהיה כבויה חצי שנייה.
 - ב. חברו את הדיודות והנגדים למיקרו-בקר והריצו את התוכנית.
- לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, שתי דיודות לד ושני נגדים של $\Omega 330$.

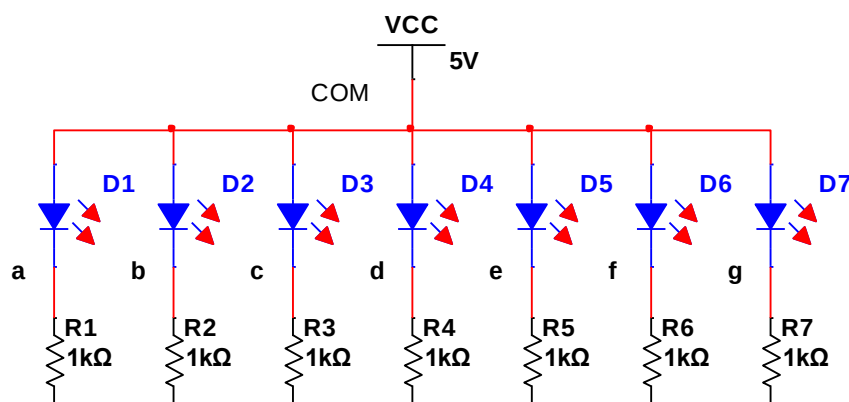
תצוגת 7 מקטעים 7 SEGMENT

תצוגת 7 מקטעים זהו מערך של 7 דיודות LED המסודרות באופן כזה שניתן להציג באמצעותן את הספרות מ-0 עד 9. ניתן לחבר את הדיודות בשני אופנים: קטודה משותפת (CC) או אנודה משותפת (CA).



הפעלת תצוגה מסוג CA תתבצע ברמה לוגית נמוכה (כמו ב- Pull-Down) ואילו תצוגה מסוג CC תתבצע ברמה לוגית גבוהה (כמו ב- Pull-Up). לכל דיודה נחבר נגד טורי על מנת לקבוע את עצמת הזרם הדרושה.

באיור הבא מתואר אופן החיבור של יחידת התצוגה:



איור 2.9 תרשים של יחידת תצוגה מסוג אנודה משותפת

כדי להציג את הספרות מ-0 עד 9 נצטרך להדליק את דיודות הLED המתאימות ביחידת התצוגה.
בטבלה שלהלן פירוט הספרות ומצב כל דיודת LED.

מספר/הדק בתצוגה	a	b	c	d	e	f	g
0	On	On	On	On	On	On	Off
1	Off	On	On	Off	Off	Off	Off
2	On	On	Off	On	On	Off	On
3	On	On	On	On	Off	Off	On
4	Off	On	On	Off	Off	On	On
5	On	Off	On	On	Off	On	On
6	On	Off	On	On	On	On	On
7	On	On	On	Off	Off	Off	Off
8	On	On	On	On	On	On	On
9	On	On	On	On	Off	On	On

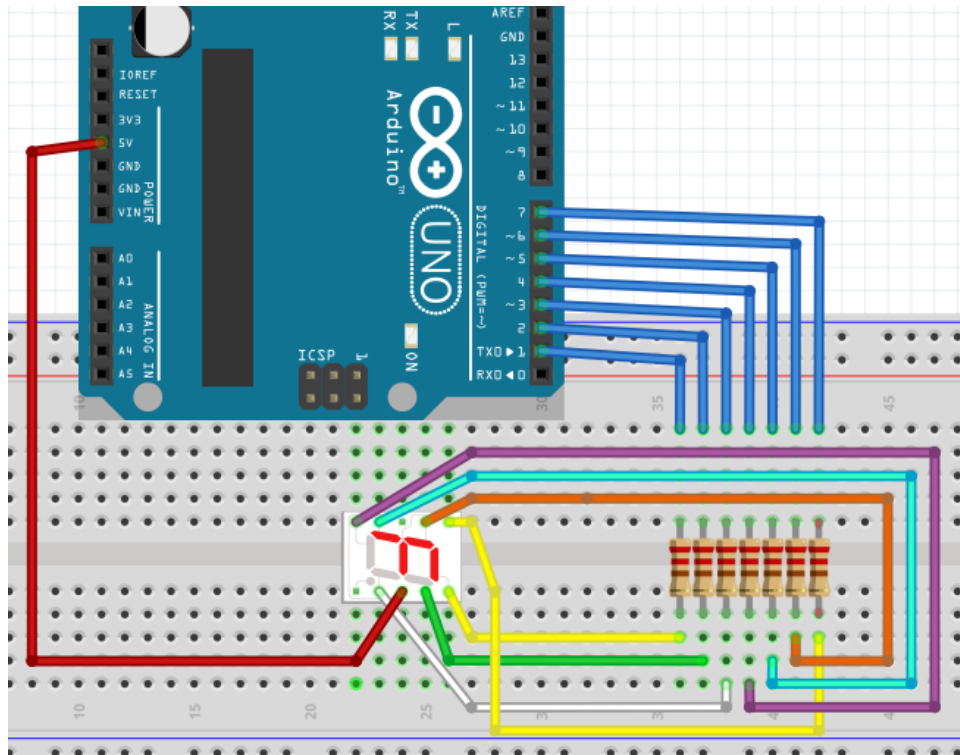
טבלה 2.4 טבלת מצבי דיודות הLED לפי ספרות

On – דיודת הLED דולקת

Off – דיודת הLED כבויה

דוגמה 2.4 – חיבור תצוגת 7-SEGMENT מסוג אנודה משותפת CA

באיור שלהלן מתואר חיבור תצוגה ללוח UNO.



איור 2.10 דיאגרמת חיבורים של יחידת תצוגה

מיפוי הדקים:

הדק בתצוגה	a	B	c	d	e	f	g
הדק בלוח UNO	1	2	3	4	5	6	7

לפי איור 2.9 המתח הדרוש כדי להדליק כל דיודת לד נמוך ממתח האספקה. מכיוון קצות הנגדים המחוברים למיקרו-בקר תיעשה ההפעלה ברמה לוגית 0 (מתח 0V בקירוב).

מצב On הוא רמה לוגית 0 (LOW) ומצב Off הוא רמה לוגית 1 (HIGH).

תוכנית להצגת המספרים 0 ו-1 לסירוגין, גרסה 1

```
void setup() {  
    pinMode(1, OUTPUT); // הגדרת הדק 1 כמוצא  
    pinMode(2, OUTPUT); // הגדרת הדק 2 כמוצא  
    pinMode(3, OUTPUT); // הגדרת הדק 3 כמוצא  
    pinMode(4, OUTPUT); // הגדרת הדק 4 כמוצא  
    pinMode(5, OUTPUT); // הגדרת הדק 5 כמוצא  
    pinMode(6, OUTPUT); // הגדרת הדק 6 כמוצא  
    pinMode(7, OUTPUT); // הגדרת הדק 7 כמוצא  
}  
  
void loop() {  
    // הספרה 0  
    digitalWrite(1, LOW);  
    digitalWrite(2, LOW);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, HIGH);  
    delay(1000); // השהיה של 1 שנייה  
    // הספרה 1  
    digitalWrite(1, HIGH);  
    digitalWrite(2, LOW);  
    digitalWrite(3, LOW);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, HIGH);  
    digitalWrite(7, HIGH);  
    delay(1000); // השהיה של 1 שנייה  
}
```

תוכנית להצגת המספרים 0 ו-1 לסירוגין, גרסה 2

```
// הגדרת קבועים עם שמות משמעותיים
const int a = 1;
const int b = 2;
const int c = 3;
const int d = 4;
const int e = 5;
const int f = 6;
const int g = 7;

void setup() {
    pinMode(a, OUTPUT); // הגדרת הדק 1 כמוצא
    pinMode(b, OUTPUT); // הגדרת הדק 2 כמוצא
    pinMode(c, OUTPUT); // הגדרת הדק 3 כמוצא
    pinMode(d, OUTPUT); // הגדרת הדק 4 כמוצא
    pinMode(e, OUTPUT); // הגדרת הדק 5 כמוצא
    pinMode(f, OUTPUT); // הגדרת הדק 6 כמוצא
    pinMode(g, OUTPUT); // הגדרת הדק 7 כמוצא
}

void loop() {
    // הספרה 0
    digitalWrite(a, LOW);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
    delay(1000); // השהיה של 1 שנייה
    // הספרה 1
    digitalWrite(a, HIGH);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, HIGH);
    digitalWrite(e, HIGH);
    digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
    delay(1000); // השהיה של 1 שנייה
}
```

מומלץ להשתמש בערכים קבועים (constants) משתי סיבות עיקריות:

1. התוכנית קריאה יותר והדבר מקנה לה משמעות שעוזרת להבין את מטרתה.
2. אפשר כך בקלות למפות מחדש את הדקי המיקרו-בקר בשינוי מקום אחד בתוכנית.

הגדרת מפתח כפלט

לעיתים נוח לפנות למפתח שלם, כמו בתצוגת שבעה מקטעים, שבה משנים בו בזמן שבעה הדקים. כפי שאפשר לראות בטבלה 2.1, הדקים 0–7 בלוח ה-UNO ממופים למפתח D - PORTD. אפשר להגדיר בו בזמן את ההדקים ככניסה או כציאה ולפנות אליהם כדי לקרוא את ערכם או לשנות אותו. לשם כך מוגדרים שלושה אוגרים (תאי זיכרון במיקרו-בקר) המאפשרים פנייה להדקים.

DDRD – מלת בקרה המאפשרת להגדיר לכל הדק מפתח D אם הוא קלט או פלט. 0 מגדיר הדק ככניסה (INPUT) ו-1 מגדיר הדק כציאה (OUTPUT).

בדוגמה שלהלן מוגדר הדק 0 בלוח הממופה להדק PD0 במיקרו-בקר ככניסה, וכל ההדקים מ-1 עד 7 הממופים להדקים PD1-PD7 מוגדרים במיקרו-בקר כציאות. האות B בתחילת הצירוף מציינת שהערך המספרי נתון לפי בסיס 2.

```
void setup() {
    DDRD = B11111110;
}
```

PORTD – אוגר במיקרו-בקר המאפשר לקרוא ולכתוב נתון לכל הדקי מפתח D בבת אחת.

PIND – אוגר לקריאה בלבד, המשקף את מצב הכניסות במפתח D.

בדוגמה שלהלן מוציאים בו בזמן 1 בהדקים PD4-PD7 ואפס בהדקים PD3-PD0. רק ההדקים שהוגדרו יציאות יושפעו מהוצאת ערכים למפתח.

```
void loop() {
    PORTD = B11110000;
}
```

דוגמה 2.5 – חיבור תצוגת SEGMENT-7 מסוג אנודה משותפת CA, גרסה 3

כתוב תוכנית להצגת המספרים 0 ו-1 בשימוש בפנייה למפתח על פי דיאגרמת החיבורים של דוגמה 4.

```
void setup() {  
    DDRD = B11111110;           // הגדרת 1-7 כיציאות 0 ככניסה  
}  
void loop() {  
    PORTD = B10000000;          // הספרה 0  
    delay(1000);                // השהיה של 1 שנייה  
    PORTD = B11110010;          // הספרה 1  
    delay(1000);                // השהיה של 1 שנייה  
}
```

שימו לב עד כמה תוכנית זו קצרה יחסית לגרסאות הקודמות.

 תרגול

שאלה 2.5

- א.** כתבו תוכנית המציגה בתצוגת שבעה מקטעים את המספרים מ-0 עד 9 כך שכל מספר יוצג חצי שנייה.
- ב.** חברו רכיב תצוגה על פי איור 2.9 ללוח UNO ובדקו את התוכנית שכתבתם. לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, יחידת תצוגה שבעה מקטעים מסוג CA ושבעה נגדים של 330Ω .

דוגמה 2.6 – הפעלת תצוגה מסוג אנודה משותפת CA, שימוש במערך

כתבו תוכנית להצגת המספרים מ-0 עד 9 באופן מחזורי תוך שימוש בפנייה למפתח ובמערך חד-ממדי על פי דיאגרמת החיבורים של דוגמה 4.

```
//
//          0          1          2          3          4
byte table7Seg[] = {B1000000, B1111001, B0100100, B0110000, B0011001,
//          5          6          7          8          9
//          B0010010, B0000010, B1111000, B0000000, B0010000};

byte index;

void setup() {
    DDRD = B11111110; // כיציאות 0 ככניסה
    index = 0;        // אתחול משתנה האינדקס למערך
}

void loop() {
    PORTD = table7Seg[index]; // הפעלת יחידת התצוגה עם הערך מהמערך
    delay(1000);              // השהיה של 1 שנייה
    index++;                  // קידום אינדקס המערך
    if (index == 10)          // בדיקה אם סיימנו עם כל איברי הנערך
        index = 0;           // אם סיימנו, חוזרים לתחילת המערך למחזור הצגה נוסף
}
```

במערך החד-ממדי table7Seg עשרה ערכים המייצגים את הספרות מ-0 עד 9, כך שבמקום 0 מופיע הערך הבינארי של הספרה 0, במקום 1 מופיע הערך של הספרה 1 וכן הלאה.

 תרגול

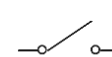
שאלה 2.6

- א. כתבו תוכנית המציגה בתצוגת שבעה מקטעים את המספרים מ-9 עד 0 בסדר יורד כך שכל מספר יוצג חצי שנייה.
- ב. חברו רכיב תצוגה על פי איור 2.9 ללוח UNO ובדקו את התוכנית שכתבתם.

2.5 מפתחי קלט

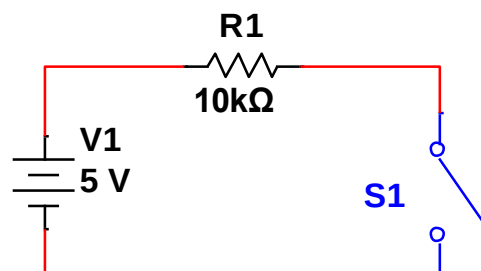
אל מפתחי הקלט נחבר התקנים ספרתיים המספקים רמות לוגיות 1 או 0, 5V או 0V בהתאמה. התקן הקלט הפשוט ביותר הוא מפסק.

מפסק הוא התקן אלקטרו-מכני הכולל שני מגעים חשמליים שנסגרים ונפתחים על ידי שינוי מצבם. יש מפסקים שיכולים להיות באחד משני המצבים באופן קבוע ויש כאלה שנמצאים במצב אחד באופן קבוע ובמצב שני באופן זמני – לחצן.

		
מפסק דו-מצבי לחיצה N.C	מפסק דו-מצבי לחיצה N.O	מפסק דו-מצבי סטטי

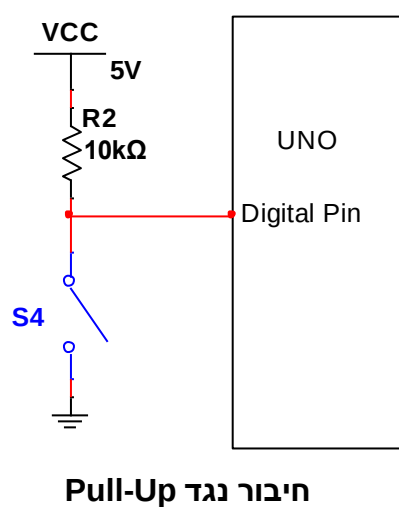
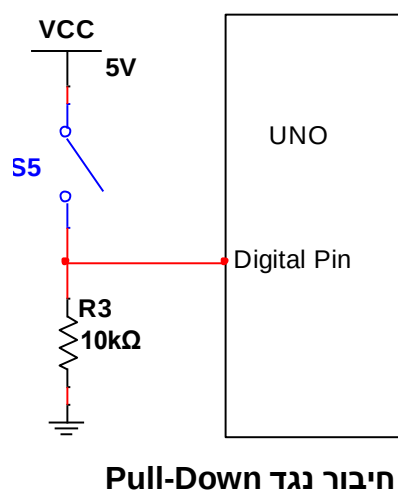
כדי "לקרוא" את מצבו של מפסק נחבר אותו למקור מתח כך שיופקו שתי רמות מתח: רמת מתח אחת למצב אחד ורמת מתח אחרת למצב השני.

המעגל הבסיסי כולל מקור מתח, נגד ומפסק:



- כאשר המפסק פתוח המתח על הנגד הוא 0V ובין הדקי המפסק יימדד מתח 5V.
- כאשר המפסק סגור המתח על הנגד הוא 5V ובין הדקי המפסק יימדד מתח 0V.

תצורות חיבור



חיבור נגד Pull-Up

בחיבור זה, כאשר המפסק פתוח יימדד מתח של 5V וכאשר המפסק סגור יימדד במפסק בהדק מתח של 0V.

חיבור נגד Pull-Down

בחיבור זה, כאשר המפסק פתוח יימדד מתח של 0V וכאשר המפסק סגור יימדד במפסק בהדק מתח של 5V.

הגדרת הדק מבוא במיקרו-בקר

לפני שימוש בהדק ספרתי יש להגדיר את אופן פעולתו: כניסה או יציאה. הגדרת ההדק נעשית בהוראה pinMode:



pinMode(pin, mode)

pin מספר ההדק הספרתי מ-0 עד 13

mode הגדרת אופן הפעולה של ההדק:

OUTPUT - מוצא

INPUT - מבוא

INPUT PULLUP – מבוא עם נגד Pull

דוגמה 2.7 – הדלקת נורת לד לפי מצב מפסק

כתוב תוכנית המפעילה את דיודת הלד המחוברת להדק 13 כאשר מפסק המחובר בתצורת Pull-Up מחובר להדק 12. כאשר המפסק פתוח נורת הלד תדלוק, ולהפך.

```
const int pinLed = 13;           // הגדרת קבוע עבור הדק הלד
const int pinSwitch = 12;        // הגדרת קבוע עבור הדק המפסק
int value;                       // הגדרת משתנה לקליטת מצב המפסק

void setup() {
    pinMode(pinLed, OUTPUT);      // הגדרת הדק הלד כמוצא
    pinMode(pinSwitch, INPUT);    // הגדרת הדק המפסק כמבוא
}

void loop() {
    value = digitalRead(pinSwitch); // קריאת מצב המפסק למשתנה
    digitalWrite(pinLed, value);     // כתיבת ערך המפסק לדיודת הלד
}
```



שאלה 2.7

- א. כתבו תוכנית הקוראת את מצב המפסק. אם המפסק סגור נורת הLED תדלוק, ולהפך.
- ב. חברו מפסק בתצורת Pull-Up להדק 12 בלוח UNO ובדקו את התוכנית שכתבתם. הוסיפו לתוכנית הוראות להצגת מצב התוכנית בשימוש בהוראות Serial.print או Serial.println.
- לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, מפסק ונגד $10k\Omega$.

שאלה 2.8

מחליפים את שתי ההוראות בפונקציית ה-loop() בדוגמה 7 בהוראה שלהלן:

```
digitalWrite(pinLed, digitalRead(pinSwitch));
```

האם התוכנית תשתנה? נמק.

דוגמה 2.8 – הבהוב נורת לד בקצב לפי מצב מפסק

כתבו תוכנית הגורמת להבהוב של דיודת הלד המחוברת להדק 13. זמן ההפעלה והכיבוי של נורת הלד יהיה תלוי במצב המפסק המחובר בתצורת Pull-Down להדק 12. כאשר המפסק פתוח נורת תהבהב בקצב של שנייה וכאשר המפסק סגור נורת הלד תהבהב בקצב של חצי שנייה.

```
const int pinLed = 13;           // הגדרת קבוע עבור הדק הלד
const int pinSwitch = 12;        // הגדרת קבוע עבור הדק המפסק
int value;                       // הגדרת משתנה לקליטת מצב המפסק
int timedelay;                   // הגדרת משתנה לקביעת זמן ההפעלה והכיבוי
void setup() {
    pinMode(pinLed, OUTPUT);      // הגדרת הדק הלד כמוצא
    pinMode(pinSwitch, INPUT);    // הגדרת הדק המפסק כמבוא
}
void loop() {
    value = digitalRead(pinSwitch); // קריאת מצב המפסק למשתנה
    if (value == HIGH)
        timedelay = 500;          // מפסק סגור
    else
        timedelay = 1000;          // מפסק פתוח
    digitalWrite(pinLed, HIGH);    // הדלקת דיודת הלד
    delay(timedelay);
    digitalWrite(pinLed, LOW);     // כיבוי דיודת הלד
    delay(timedelay);
}
```

שאלה 2.9

א. כתבו תוכנית הקוראת את מצבם של שני מפסקים המחוברים בתצורת Pull-Up. המפסקים מחוברים להדקים 10 ו-11. נורת לד המחוברת להדק 13 תהבהב בקצב התלוי במצבי המפסקים, כמפורט בטבלה שלהלן.

מצב מפסק 1	מצב מפסק 2	קצב הבהוב (mS)
פתוח	פתוח	250
סגור	פתוח	500
פתוח	סגור	750
סגור	סגור	1000

ב. חברו את המפסקים ללוח UNO ובדקו את התוכנית שכתבתם. הוסיפו לתוכנית הוראות להצגת מצב התוכנית בשימוש בהוראות Serial.print או Serial.println. לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, שני מפסקים ושני נגדים של $10k\Omega$.

שאלה 2.10

א. שני מפסקים מחוברים בתצורת Pull-Up להדקים 10 ו-11. תצוגת שבעה מקטעים מחוברת על פי איור 2.9.
a. כתבו תוכנית הקוראת את מצב המפסקים ומציגה את המספרים:

ערך מפסק 1	ערך מפסק 2	ערך תצוגה
0	0	4
1	0	5
0	1	6
1	1	7

ב. חברו את המפסקים ואת תצוגת שבעת המקטעים ללוח UNO ובדקו את התוכנית שכתבתם. הוסיפו לתוכנית הוראות להצגת מצבה בשימוש בהוראות Serial.print או Serial.println. לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, שני מפסקים, שני נגדים של $10k\Omega$, תצוגת שבעה מקטעים מסוג CA ושבעה נגדים של $\Omega 330$.

קריאת מפתחים

בדוגמאות הקודמות ראינו שימוש במילות בקרה המתייחסות למפתחי המיקרו-בקר:

- DDRB, DDRC, DDRD – הגדרות אופני עבודה של מפתחי קלט/פלט ספרתיים;
- PORTB, PORTC, PORTD – פנייה לקריאה ממפתח או כתיבה למפתח;
- PINB, PINC, PIND – פנייה לקריאה בלבד במפתחים.

כאמור, הפנייה למפתח הוא לקבוצה של שמונה הדקים. קריאת מפתח מעבירה מידע של בייט (byte). כידוע, בייט היא יחידת מידע בת שמונה סיביות.

למשל, ההוראה $value = PINB$; תקרא מידע בגודל בייט ממפתח B.

7	6	5	4	3	2	1	0
X	X	13	12	11	10	9	8

ערך משתנה value

X מצוין ערך לא ידוע

13–8 מייצג את הערכים בהדקים אלו

מיסוך סיביות

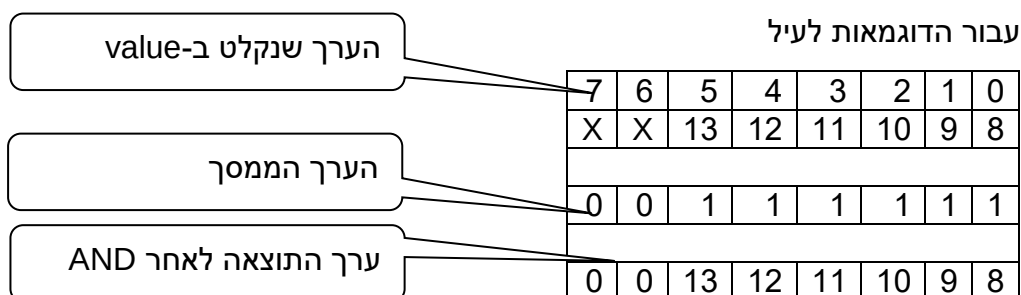
כדי לבודד קבוצת סיביות מתוך השמונה, נשתמש בפעולת AND לוגי. פעולה זו מסומנת בשפת C ונגזרותיה בסימן &.

כדי לבודד את קבוצת הסיביות נגדיר ערך בינארי המכיל ערך 1 כנגד כל סיבית רצויה וערך 0 כנגד סיבית לא רצויה.

למשל, הערך שלהלן ימסך את הסיביות 0–5:

0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

לאחר מכן נבצע פעולת AND בין המידע שהתקבל ובין הערך הבינארי הרצוי.



למשל, ההוראה `pinsValue = value & B00111111;` תציב במשתנה `pinsValue` את תוצאת פעולת המיסוך.

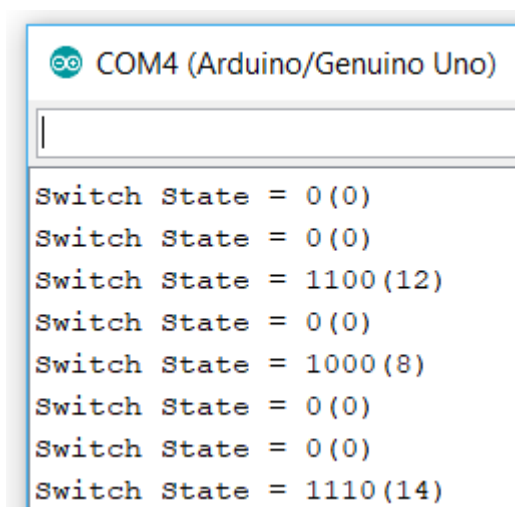
דוגמה 2.9 – קריאת קבוצת מפסקים בהדקים 11–8

מחברים ארבעה מפסקים מסוג NO למבואות 11–8 במיקרו-בקר.

```
byte value; // הגדרת משתנה לקליטת מצב ההדקים
byte switches; // הגדרת משתנה לאחסון מידע המפסקים
void setup() {
    DDRB = DDRB & B11110000; // הגדרת הדקים כקלט
    // ללא השפעה על ההדקים האחרים

    Serial.begin(9600);
}
void loop() {
    value = PINB; // קריאת מצב המפתח למשתנה
    switches = value & B00001111; // מיסוך 4 המפסקים
    Serial.print("Switch State = "); // הצגת מידע טקסטואלי
    Serial.print(switches, BIN); // הצגת ערך בינארי
    Serial.print("("); // הוספת פותח
    Serial.print(switches); // הצגת ערך עשרוני
    Serial.println(")"); // סוגר ומעבר שורה
}
```

פלט לדוגמה,



```
COM4 (Arduino/Genuino Uno)

Switch State = 0(0)
Switch State = 0(0)
Switch State = 1100(12)
Switch State = 0(0)
Switch State = 1000(8)
Switch State = 0(0)
Switch State = 0(0)
Switch State = 1110(14)
```

שאלות חזרה

שאלה 2.11

מחברים מפסק לחיצה להדק 11 ונורת לד להדק 13 במיקרו-בקר. כל לחיצה על המפסק תגרום לנורת הלד להחליף את מצבה. אם הנורה פעלה לפני הלחיצה היא תכבה, ואם הייתה כבויה היא תידלק.

- א. שרטטו סכמת חיבורים למיקרו-בקר.
- ב. הגדירו את אופני הפעולה של הדקים 11 ו-13.
- ג. כתבו תוכנית המתאימה לדרישה.
- ד. הריצו את התוכנית ובדקו אותה. הוסיפו לתוכנית הוראות להצגת מצב המפסק ונורת הלד בשימוש בהוראות `Serial.print` או `Serial.println`.

שאלה 2.12

מחברים מפסק לחיצה להדק 11 ושתי נורות לד להדקים 12 ו-13 במיקרו-בקר. כל לחיצה על המפסק תגרום לנורת לד אחת להידלק ולאחרת להיות במצב כבוי. שרטטו סכמת חיבורים למיקרו-בקר.

- א. הגדירו את אופני הפעולה של הדקים 11, 12 ו-13.
- ב. כתבו תוכנית המתאימה לדרישה.
- ג. הריצו את התוכנית ובדקו אותה. הוסיפו לתוכנית הוראות להצגת מצב המפסק ונורת הלד בשימוש בהוראות `Serial.print` או `Serial.println`.

שאלה 2.13

מחברים מפסק לחיצה להדק 0 ויחידת תצוגה של שבעה מקטעים מסוג CA להדקים 1–7 במיקרו-בקר. לחיצה ראשונה על המפסק תגרום להפעלת התצוגה כך שכל שנייה תוצג ספרה עוקבת מ-0 עד 9 באופן מחזורי. לחיצה נוספת על המפסק תעצור את התקדמות הערכים בתצוגה (פעולה הדומה לשעון עצר בעל תצוגת שנייה בלבד).

- א. הגדירו את אופני הפעולה של הדקים 0–7.
- ב. כתבו תוכנית המתאימה לדרישה.
- ג. הריצו את התוכנית ובדקו אותה.

שאלה 2.14

כתבו תוכנית המציגה ב-Serial Monitor את ערכו של הדק 10 במיקרו-בקר המוגדר הדק כניסה INPUT.

- א. מהם הערכים המתקבלים?
- ב. אחד מכם ייגע בהדק 10. האם הערכים משתנים?
- ג. מה מסקנתכם?

שאלה 2.15

כתבו תוכנית המציגה ב-Serial Monitor את ערכו של הדק 10 במיקרו-בקר המוגדר הדק כניסה INPUT_PULLUP.

- א. מהם הערכים המתקבלים?
- ב. אחד מכם ייגע בהדק 10. האם הערכים משתנים?
- ג. מה מסקנתכם?

שאלה 2.16

כתבו תוכנית המפעילה נורת לד המחוברת להדק 13 במיקרו-בקר בתדר 10Hz. זמן המחזור הוא $T = 1/f = 0.1\text{Sec}$. כלומר בחצי זמן מחזור 0.005Sec נורת הלד תדלוק ובחצי האחר היא תהיה כבויה. לאחר כל הדלקה או כיבוי הציגו ב-Serial Monitor את מצב נורת הלד תוך קריאת מצב הדק 13.

- א. מהם הערכים המתקבלים?
- ב. מה מסקנתכם?

שאלה 2.17

כתבו תוכנית המפיקה גל ריבועי בתדר $f=2\text{KHz}$ בהדק 10 במיקרו-בקר. היעזרו באלגוריתם שלהלן:

- א. הפיקו 1 בהדק 10 למשך מחצית זמן המחזור.
- ב. המתינו מחצית זמן מחזור.
- ג. הפיקו 0 בהדק 10 למשך מחצית זמן מחזור.
- ד. המתינו מחצית זמן מחזור.

כדי להמתין השתמשו בהוראה delayMicroseconds. תיעוד הוראה זו יש בקישור:

<https://www.arduino.cc/reference/en/language/functions/time/delaymicroseconds/>

בדקו באמצעות משקף תנודות את האות המתקבל בהדק 10 במיקרו-בקר.

סיכום פרק 2

בפרק 2 למדתם את הנושאים האלה:

1. הגדרת מפתח ספרתי
2. מפתחי קלט ופלט
3. הגדרת הדק כקלט או כפלט pinMode
4. הגדרת מפתח כקלט או כפלט DDRB, DDRC, DDRD
5. גישה למפתח לקריאה ולכתיבה PORTB, PORTC, PORTD
6. גישה למפתח לקריאה בלבד PINB, PINC, PIND
7. שימוש במשתנה byte
8. שימוש במשפט תנאי
9. שימוש במערך חד-ממדי
10. שימוש במסוף Serial Monitor להצגת המידע טקסטואלי בינארי ועשרוני

פרק 3: מבואות ומוצאים תקביליים

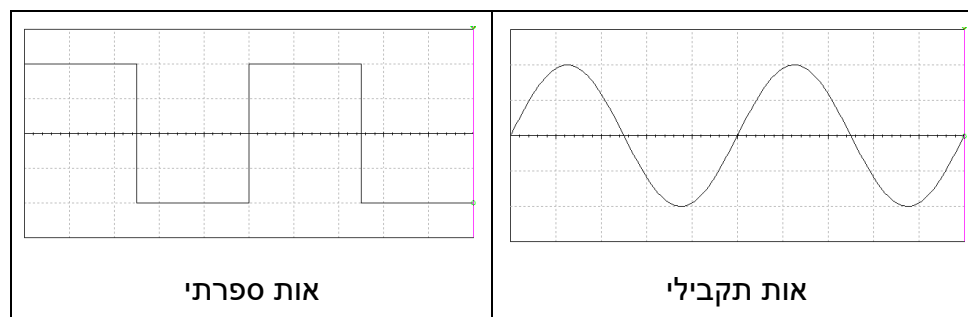
בפרק הקודם למדנו כיצד להשתמש במבואות ומוצאים ספרתיים. המאפיין החשוב של אות ספרתי הוא שתי רמות לוגיות המייצגות את ערכו, 0 או 1 (V0 או V5). המעבר בין שתי הרמות נעשה בבת אחת.

בפרק זה נתמקד ביחידות המסומנות באיור שלהלן. יחידות אלה מספקות אמצעים לחיבור התקנים אנלוגיים, בעברית: **תקביליים**.



אות תקבילי

אות תקבילי יכול לקבל כל ערך בתחום מוגדר. אחד המאפיינים הנוספים שלו הוא המשכיות בעוצמת האות, כלומר לא יופיעו באות ערכים שונים באופן משמעותי בנקודת זמן מסוימת. אין באות קפיצות מתח.



3.1 מפתח תקבילי

בפרקים הקודמים למדנו שבמערכת ממוחשבת שלושה מרכיבים יסודיים: מעבד, זיכרון והתקני קלט/פלט. תפקיד התקני הקלט/פלט הוא לחבר את ה"עולם" למעבד, כלומר להעביר מידע ממקורות חיצוניים למעבד ולהפך. כמו כן למדנו כיצד לחבר התקני קלט/פלט ספרתיים. ביחידה זו נלמד כיצד לחבר התקנים תקביליים.

דוגמה – הפעלה אוטומטית של חימום

נניח שאנו מעוניינים לשמור על טמפרטורה קבועה של מים חמים במכל בכל תנאי אפשרי.

מה דרוש לשם כך?

1. רכיב שיזהה את חום המים – חיישן טמפרטורה;
2. מערכת הפעלה לחימום – גוף חימום;
3. מערכת שתקלוט את המידע מחיישן הטמפרטורה ותפעיל את גוף החימום בעוצמה הנדרשת.



הבקר מקבל מידע מהחיישן ומבצע החלטה להפעיל חימום בעוצמה הנדרשת. **אם טמפרטורת המים נמוכה בהרבה מדרישת החימום, גוף החימום יופעל בעוצמה מרבית, וככל שהטמפרטורה קרובה לנדרש תפחת עוצמת החימום, עד להפסקת התהליך.**

אם נתייחס לבקר כאל מרכז המערכת, נזהה שהמידע מהחיישן **נקלט** על ידי הבקר, ואז מתבצע תהליך הפעלת גוף החימום (**פלט**).

- החיישן קורא מספר גדול של ערכי טמפרטורה.
- גוף החימום מופעל ברמות רבות של חימום.

בשני המקרים החיישן וגוף החימום נמצאים במצבים "מרובים".

החיישן הוא כניסה תקבילית לבקר, והמפעיל הוא יציאה תקבילית מהבקר.

כניסה אנלוגית – analog input

הדק (Pin) בבקר, שנועד לקלוט מידע מרכיב קלט בדרך כלל בתחום המתחים שבין 0V ל-5V.

יציאה אנלוגית – analog output

המונח יציאה אנלוגית במיקרו-בקר אינו מדויק מכיוון שאין בפועל יציאת מתח המספקת מתח רציף. במקום אספקת מתח רציף מסופק אות ספרתי PWM, שאת מאפייניו נכיר בהמשך הפרק הזה. מדובר בהדק בבקר, שנועד לפלוט מידע לרכיב פלט בעל שני מצבים, 0 או 1, בשיטת PWM.

3.2 מוצאים ומבואות אנלוגיים בלוח UNO

מוצאים אנלוגיים

באיור 3.1 אפשר לראות בלוח **מחבר כניסות/יציאות דיגיטליות**. איור 3.2 מראה את המחבר הזה מקרוב. במחבר מופיעים ערכים מספריים בין 0 ל-13. ההדקים 3, 5, 6, 9, 10 ו-11 מסומנים ב-"~". סימן זה מציין שאפשר להפיק במוצאים הספרתיים האלה אותות PWM המוגדרים יציאות אנלוגיות. בסך הכול יש בלוח שישה מוצאים אנלוגיים.



איור 3.2 מחבר מפתחי מוצא אנלוגיים (PWM~)

כדי להשתמש בהדקים כיציאות אנלוגיות נגדיר אותם מוצא ספרתי (OUTPUT).

מבואות אנלוגיים

באיור 3.1 אפשר לראות את מחבר הכניסות האנלוגיות. בלוח שישה מבואות אנלוגיים.



איור 3.3 תיאור מחבר מבואות אנלוגיים

3.3 מאפיינים חשמליים

מאפייני מבוא אנלוגי

כדי להשתמש בהדקים A5–A0 כהדקי מבוא אנלוגיים נועד שהם מוגדרים כניסות ספרתיות רגילות, INPUT (אין צורך להגדיר אותם כך באופן מיוחד, מכיוון שזו ברירת המחדל).

אפשר לחבר לכל אחד מהמבואות מתח בתחום 0V–5V. כל מתח בתחום זה מתורגם לערך מספרי מ-0 עד 1023 בהתאמה.

הרזולוציה – כושר ההבחנה – של הממיר היא $\Delta V = 5V/1024 = 4.88mV$.

זמן ההמרה, כלומר הזמן הדרוש לתרגום את המתח לערך מספרי, הוא $t_c = 100\mu Sec$.

$0V \leq V_{in} \leq 5V$	תחום מתחי המבוא האנלוגי
$0 \leq D \leq 1023$	תחום ערכים ספרתיים
$\Delta V = 4.88mV$	כושר הבחנה – רזולוציה
$t_c = 100\mu Sec$	זמן המרה

הנוסחה לחישוב הערך הספרתי בתלות במתח נתונה בנוסחה שלהלן:

$$D = \left[V_{in} / \Delta V \right]$$

הערך השלם של היחס בין המתח האנלוגי בכניסה לבין כושר ההבחנה (הרזולוציה).

שינוי תחום מתחי המבוא

הערכים שתוארו לעיל הם ערכי ברירת המחדל של המיקרו-בק. אפשר לשנות את הערכים האלה בהוראה `analogReference(type)`, המאפשרת לקבוע אחת משלוש אפשרויות (types):

- **DEFAULT** – מתח ייחוס פנימי 5V. ערכי מחדל, מתחי מבוא 0V עד 5V, כפי שתואר לעיל.
- **INTERNAL** – מתח ייחוס פנימי 1.1V, המרה של מתחי מבוא מ-0V עד 1.1V.
- **EXTERNAL** – מתח ייחוס חיצוני 0V עד 5V. מתחי המבוא נקבעים על פי מתח המסופק להדק AREF הסמוך להדק 13 במחבר הספרתי.

בחירת מתח ייחוס Vref משפיעה על תחום מתחי המבוא $0V \leq V_{in} \leq V_{ref}$

$$\Delta V = V_{ref} / 1024$$

כושר ההבחנה משתנה בהתאמה

 תרגול

שאלה 3.1

מפעילים את המיקרו-בקר לפי ברירת המחדל.

- א. נקרא ערך ספרתי 677. מהו המתח בכניסה האנלוגית?
- ב. מה תהיה הקריאה אם נחבר מתח של 5V בכניסה האנלוגית?

שאלה 3.2

מגדירים את מתח הייחוס בהוראה: analogReference(INTERNAL);.

- א. מהו תחום מתחי המבוא שצריך לחבר למבוא האנלוגי?
- ב. מהו כושר ההבחנה של הממיר?
- ג. מהו הערך הספרתי המתקבל אם מחברים אות מתח של 0.75V לכניסה האנלוגית?
- ד. מתקבל ערך ספרתי 768. מהו המתח במבוא הכניסה האנלוגית?

שאלה 3.3

נתון התקן אנלוגי המספק מתח מ-0V עד 2.56V. אנחנו מעוניינים לקבל קריאות ספרתיות מ-0 עד 1023 לתחום המתח הנתון.

- א. מהו מתח הייחוס הדרוש? לאן צריך לחבר אותו?
- ב. מהי ההוראה המגדירה את מתח הייחוס?
- ג. מתקבלת קריאה ספרתית של 512. מהו מתח המבוא?

מאפייני מוצאים אנלוגיים

מכיוון שכאמור, בפועל אין מוצא אנלוגי, אלא מוצא ספרתי בשיטת PWM המשמש כיציאה אנלוגית, מאפייני החשמליים של מוצא אנלוגי זהים למאפייני מוצא ספרתי. המאפיינים החשמליים של מוצא ספרתי פורטו בפרק הקודם, בסעיף 2.3.

בשיטת PWM נדון בהרחבה בסעיף הבא.

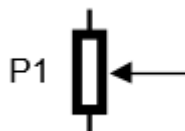
3.4 מבוא אנלוגי

נחבר למבוא אנלוגי רכיב המספק מתחים בתחום המתאים למיקרו-בקר, בדרך כלל מ-0V עד 5V.

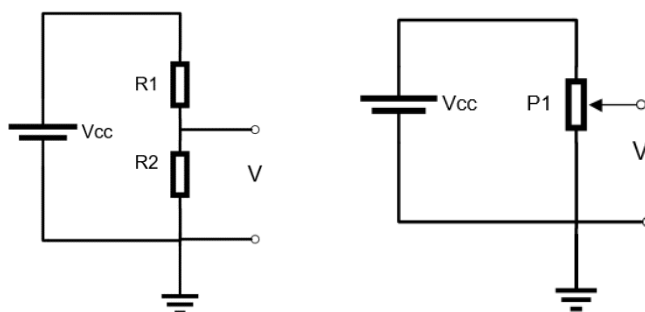
בסעיף זה נציג חיבור התקנים שונים למיקרו-בקר: נגד משתנה, חיישן טמפרטורה LM35 ונגד רגיש לאור LDR.

נגד משתנה – פוטנציומטר

נגד משתנה הוא התקן עם שלושה הדקים; שני הדקים הם קצותיו של נגד והשלישי הוא "זחלן" הנע בין קצות הנגד. בכל נקודה הזחלן מחלק את ההתנגדות הכוללת לשני נגדים: אם נחבר בין הדקי הפוטנציומטר מקור מתח נקבל בין הדקי הזחלן מתח התלוי במיקום שלו.



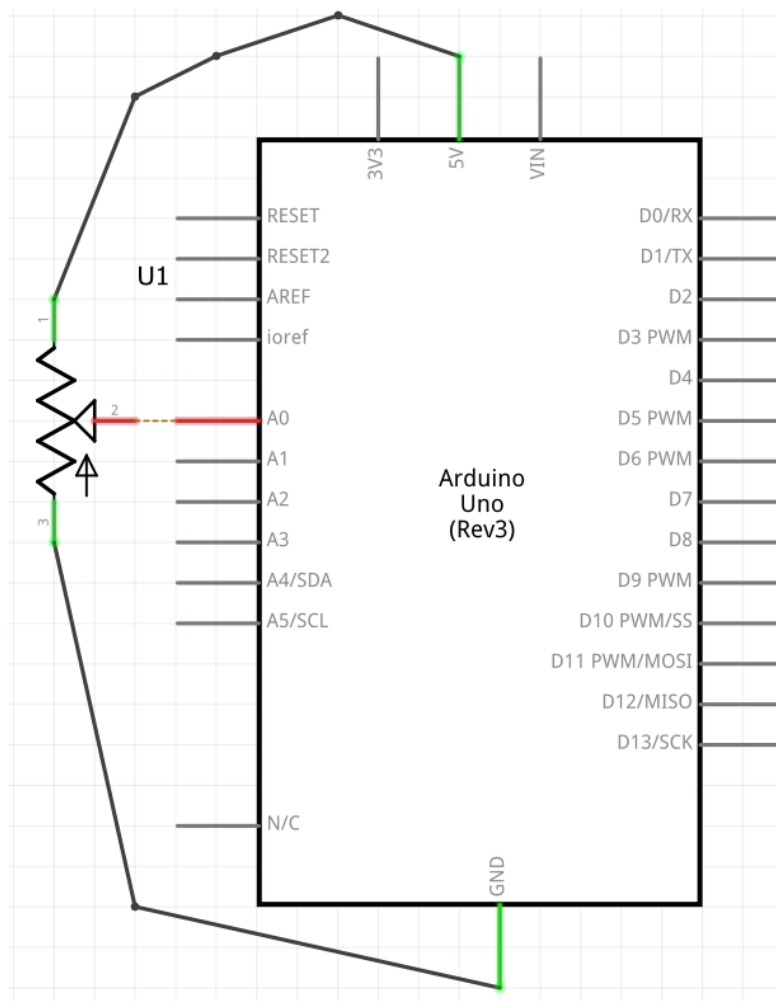
מעגל שקול הוא שני נגדים המחוברים בטור.



על פי נוסחת מחלק המתח, בין הזחלן לאדמה המתח הוא: $V = V_{cc} \frac{R_2}{R_1 + R_2}$

תחום המתח V הוא מ-0V עד V_{cc} .

חיבור פוטנציומטר למיקרו-בקר



איור 3.4 חיבור פוטנציומטר לכניסה אנלוגית A0

קריאת הערך האנלוגי

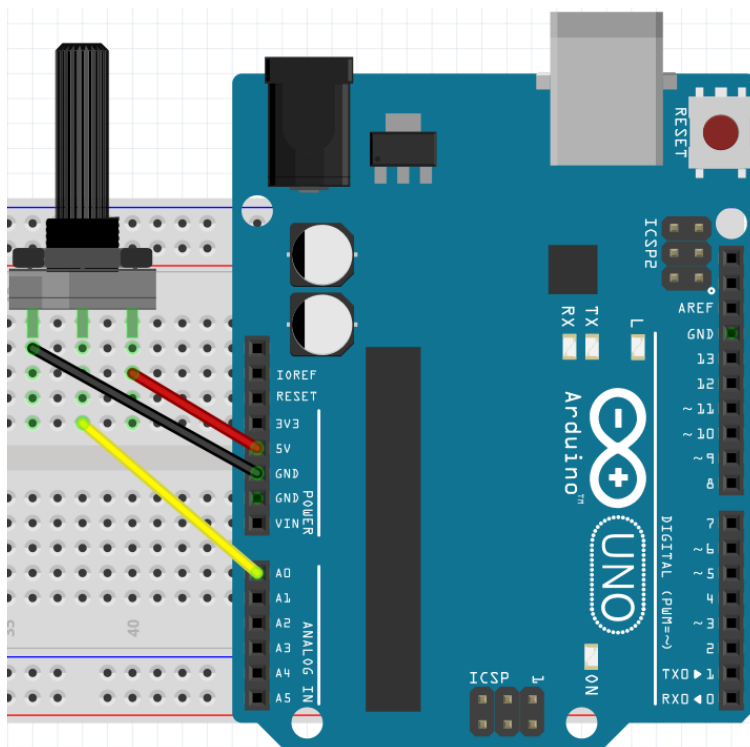
ההוראה `analogRead(pin)` משמשת לקריאת הערך האנלוגי מההדק המתאים. ההוראה מחזירה ערך מספרי שלם בין 0 ל-1023 בתלות במתח הכניסה.

למשל, ההוראה `value = analogRead(A0);` תקרא את ערך המתח בהדק A0 ותחזיר ערך מספרי מתאים.

השימוש בערכים 0–5 כפרמטר שקולים לשימוש ב-A0 עד A5 בהתאמה.

דוגמה 3.1 – קריאת מצב פוטנציומטר מכניסה אנלוגית A0

נחבר פוטנציומטר בקצותיו למתח 5V וזחלן למבוא אנלוגי.



התוכנית לקריאת ערך המתח:

```
int analogPin = A0;           // הדק המבוא האנלוגי שהזחלן מחובר אליו
int val = 0;                   // משתנה לקריאת ערך האות האנלוגי
void setup() {
  Serial.begin(9600);          // אתחול תקשורת
}
void loop() {
  val = analogRead(analogPin); // קריאת הערך האנלוגי
  Serial.println(val);          // הצגת הערך האנלוגי
}
```

דוגמה 3.2 – קריאת מצב פוטנציומטר מכניסה אנלוגית A0 וחישוב המתח

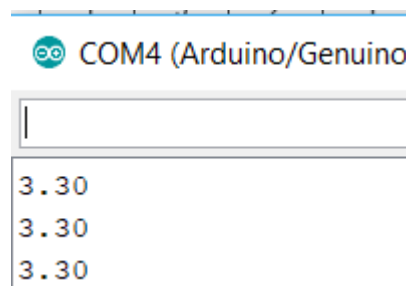
כמו בדוגמה הקודמת, תוכנית זו תקרא את ערך המתח ובמקום להציג את ערכו המספרי השלם בין 0 ל-1023 יוצג ערך המתח בהדק.

```
int analogPin = A0; ;           // הדק המבוא האנלוגי שהזחלן מחובר אליו
int val = 0; ;                  // משתנה לקריאת ערך האות האנלוגי
float voltage;                  // ערך המתח לאחר חישוב
void setup() {
    Serial.begin(9600); // אתחול תקשורת
}
void loop() {
    val = analogRead(analogPin); // קריאת הערך האנלוגי
    voltage = 0.00488 * val; // חישוב ערך המתח לפי הערך הספרתי
    Serial.println(voltage); // הצגת ערך המתח
}
```

הסבר החישוב

הנוסחה לחישוב הערך הספרתי על פי המתח $D = \left[V_{in} / \Delta V \right]$. כדי לקבל את ערך המתח נכפול את המשוואה בכושר ההבחנה ΔV : $V_{in} = \Delta V \times D$.

תוצאת הרצה עבור מתח של 3.3V בכניסה האנלוגית.





שאלה 3.4

זחלן של פוטנציומטר מחובר להדק A2 במיקרו-בקר. מתח הייחוס נקבע כפנימי INTERNAL.

- א. מה ערך מתח המבוא המרבי שיש לו משמעות?
 - ב. כתבו תוכנית הקוראת את ערך המתח במבוא ומציגה אותו.
 - ג. חברו פוטנציומטר ללוח ובדקו את התוכנית.
- לצורך הבדיקה הצטיידו בלוח UNO, כבל תקשורת, חוטי חיבור, ופוטנציומטר של $10k\Omega$.

שאלה 3.5

כפי שכבר צוין בסעיף 3.3 מאפיינים חשמליים, זמן ההמרה הנדרש לרכיב הוא כ- $100\mu\text{Sec}$ מרגע הפעלת הפעולה עד לקבלת הערך.

- א. מדוע אין בדוגמאות 3.1 ו-3.2 התייחסות לזמן ההמרה?
- ב. אם נשמיט את השורה `Serial.println(val);` מדוגמה 3.1, האם עדיין תהיה התוכנית תקינה? הסבירו.

שאלה 3.6

כדי לדייק בקריאת ערכים אנלוגיים ולהפחית במידה רבה את ההשפעות של "רעש" במערכת, מקובל לבצע כמה קריאות ולחשב את הממוצע שלהן. כתבו תוכנית על פי דוגמה 3.1 המחשבת בכל פעם ממוצע של עשר קריאות עוקבות ומציגה את הממוצע. היעזרו באלגוריתם שלהלן:

1. אפסו משתנה סכום `sum=0`.
2. חזרו עשר פעמים.
- א. קראו ערך אנלוגי למשתנה `Val`.
- ב. חברו את סכום המשתנה `sum` עם המשתנה `Val`.
3. חשבו את הממוצע `sum/10`.
4. הציגו את ערך הממוצע.

שימו לב!

בין קריאה לקריאה יש להמתין `(delayMicroseconds())`, זמן השקול לזמן המרה.

נגד רגיש לאור

נגד רגיש לאור (LDR - Light Depended Resistor) הוא רכיב המשנה את התנגדותו לפי עוצמת האור ולפי אורך הגל של האור (צבע האור). בהנחה שהאור שלנו כולל את כל אורכי הגל נוכל להתייחס רק לעוצמת ההארה כגורם המשנה את ההתנגדות.

כידוע, ההתנגדות של נגד תלויה באורכו, בשטח החתך ובהתנגדות הסגולית שלו. ההתנגדות של ה-LDR תלויה רק בהתנגדות הסגולית של הנגד, וזו מושפעת משינוי עוצמת ההארה.

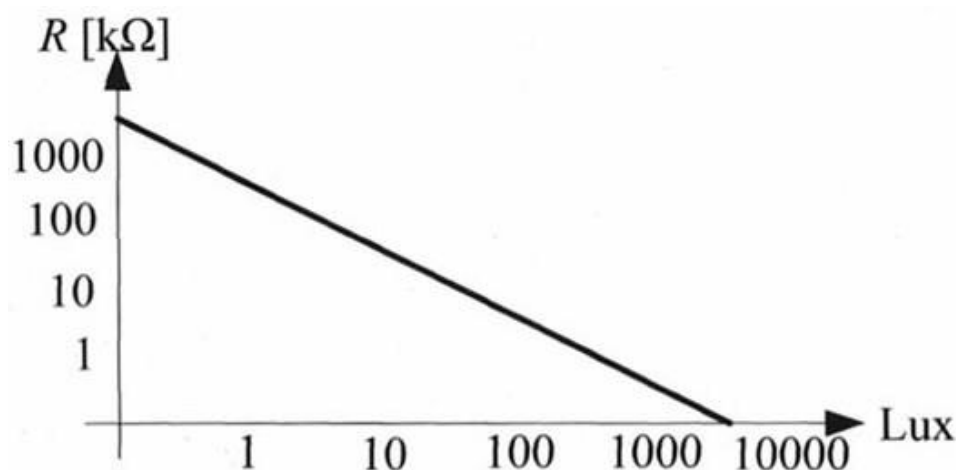
יש יחס הפוך בין עוצמת האור לבין ההתנגדות; ככל שעוצמת האור גבוהה יותר התנגדות הנגד נמוכה יותר, ולהפך. הארה של $10 \div 10000$ לוקס (Lux) משתנה מאומים בודדים עד עשרות $k\Omega$.

בטבלה שלפניכם מתוארים ערכים אופייניים של שני נגדים הרגישים לעוצמת הארה.

סוג הרכיב	התנגדות אופיינית בחשכה	התנגדות בתאורה 10 lux	התנגדות בתאורה 100 lux
NOPR12	1 M Ω	9 k Ω	400 Ω
NSL-M51	20 M Ω	(20 ÷ 100) k Ω	5 k Ω

טבלה 3.1 ערכים אופייניים של נגדים הרגישים לאור

באיור 3.5 מוצג גרף המתאר את התנגדות הנגד כפונקציה של עוצמת ההארה (גרף לוגריתמי).

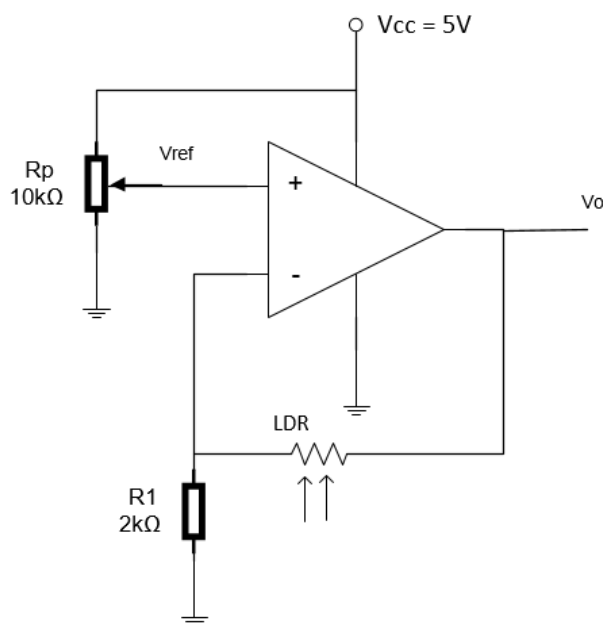


איור 3.5 התנגדות LDR ביחס לעוצמת הארה

מדידות עוצמת הארה

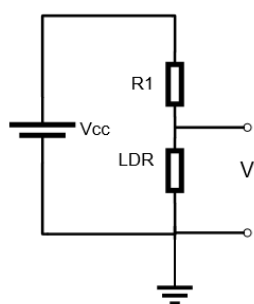
מגבר שרת בחיבור לא הופך

כדי למדוד את עוצמת ההארה באמצעות LDR מודדים את המתח ממגבר שרת בעל נגד LDR המחובר כנגד משוב, בתצורת מגבר לא מהפך, כמתואר באיור 3.6



איור 3.6 חיבור חיישן LDR במעגל מגבר לא מהפך

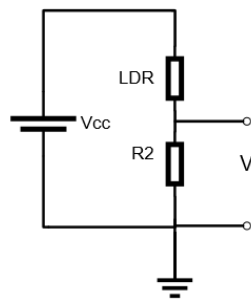
מחלקי מתח - חיבור LDR ביחס לאדמה



ככל שעוצמת האור גבוהה יותר כך המתח נמוך יותר, ולהיפך.

$$V = V_{cc} \frac{LDR}{LDR + R_2}$$

מחלקי מתח – חיבור LDR יחס ל-

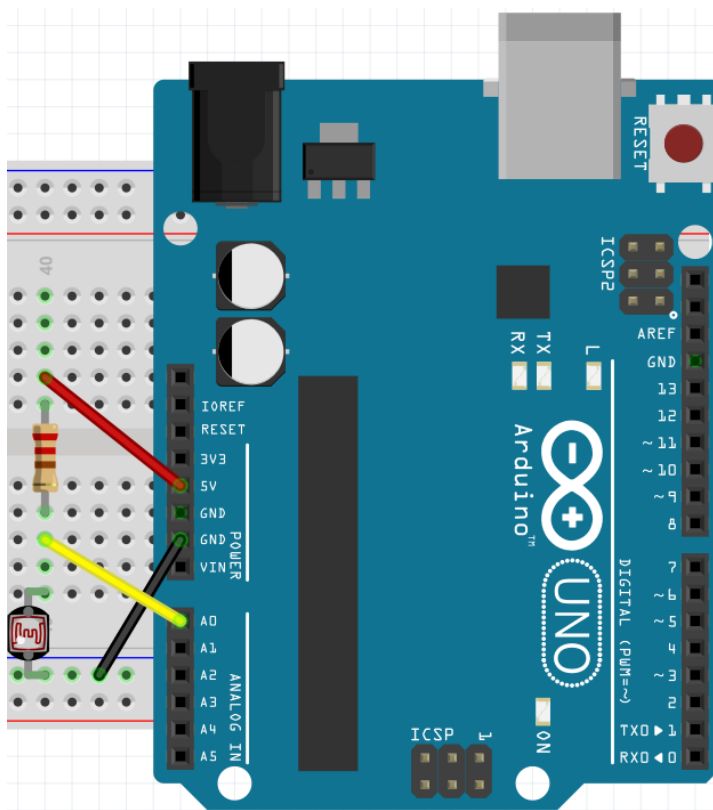


ככל שעוצמת האור גבוהה יותר כך המתח גבוה יותר, ולהיפך.

$$V = V_{cc} \frac{R_2}{LDR + R_2}$$

דוגמה 3.3 – חיבור LDR למבוא אנלוגי – זיהוי סף תאורה

באיור שלהלן נראה חיבור LDR ללוח UNO.



איור 3.7 דיאגרמת חיבורים של LDR ביחס לאדמה

בדוגמה זו, ככל שעוצמת האור גדולה יותר המתח נמוך יותר, ולהפך: ככל שעוצמת האור נמוכה יותר המתח גדול יותר.

אם המתח המתקבל במבוא גבוה מ-4V (עוצמת הארה נמוכה), תידלק נורת הלד. אם המתח המתקבל נמוך מ-3V (עוצמת הארה גבוהה) נורת הלד תכבה. בתחום שבין 3V ל-4V לא יהיה שינוי במצב של נורת הלד.

```
int analogPin = A0; //הדק המבוא האנלוגי שחיישן האור מחובר אליו
int ledPin = 13;      //הדק נורת הLED
int val = 0;          //הערך הספרתי של החיישן
float voltage;         //ערך המתח התואם לערך הספרתי
void setup() {
    Serial.begin(9600); //אתחול תקשורת טורית
    pinMode(ledPin, OUTPUT); //הגדרת הדק הנורה כמוצא
    digitalWrite(ledPin, LOW); //כיבוי הנורה
}
void loop() {
    val = analogRead(analogPin); //קריאת הערך הספרתי של חיישן האור
    voltage = 0.00488 * val;      //חישוב המתח בחיישן האור
    Serial.println(voltage);      //הצגת ערך מתח של חיישן האור
    if (voltage > 4.0)            //בדיקת סף מתח עליון
        digitalWrite(ledPin, HIGH); //הדלקת הנורה
    else if (voltage < 3.0)       //בדיקת סף מתח תחתון
        digitalWrite(ledPin, LOW);  //כיבוי הנורה
}
```

 תרגול

שאלה 3.7

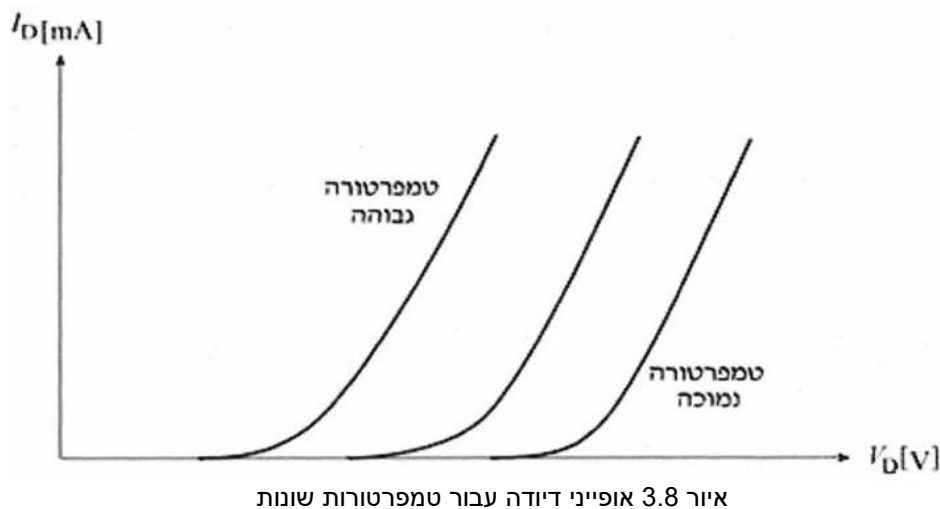
א. חברו חיישן אור LDR לכניסה אנלוגית A2 וארבע נורות LED להדקים 8, 9, 10 ו-11. הדליקו את הנורות לפי הטבלה:

LED4(11)	LED3(10)	LED2(9)	LED1 (8)	V_{LDR}
Off	Off	Off	On	$0V < V_{LDR} < 1V$
Off	Off	On	On	$1V < V_{LDR} \leq 2V$
Off	On	On	On	$2V < V_{LDR} \leq 3.5V$
On	On	On	On	$3.5V < V_{LDR}$

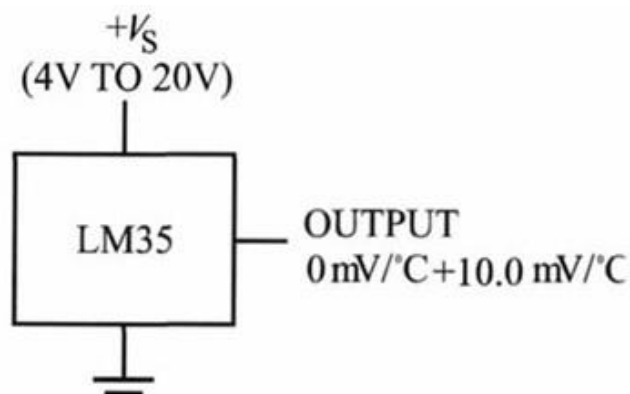
- ב.** שרטטו מעגל חשמלי לחיבור החיישן ונורות הLED ללוח ה-UNO.
- ג.** כתבו תוכנית המתאימה למעגל שתכנתם בסעיף הקודם, שתקרא את הערך של חיישן האור ותדליק את נורות הLED לפי הדרישה בטבלה.
- ד.** חברו את הרכיבים על פי המעגל החשמלי ובדקו את התוכנית שכתבתם.

חיישן טמפרטורה LM35

חיישן הטמפרטורה LM35 הוא רכיב מוכלל דיודה רגישה לשינויי טמפרטורת הסביבה. כאשר הדיודה מחוברת בממתח קדמי ומזרימים דרכה זרם קבוע, המתח על הדיודה קטן בגודל אופייני של 2mV עם עלייה של מעלת צלסיוס אחת. באיור 3.8 מתוארים אופיינים קדמיים של דיודה עבור טמפרטורות שונות. אפשר להבחין שככל שהטמפרטורה גבוהה יותר כך מפל המתח על הדיודה V_D קטן עבור אותו זרם. התלות של המתח במוצא החיישן בטמפרטורה, היא קווית (ליניארית) ועומדת על 10mV למעלת צלסיוס אחת.



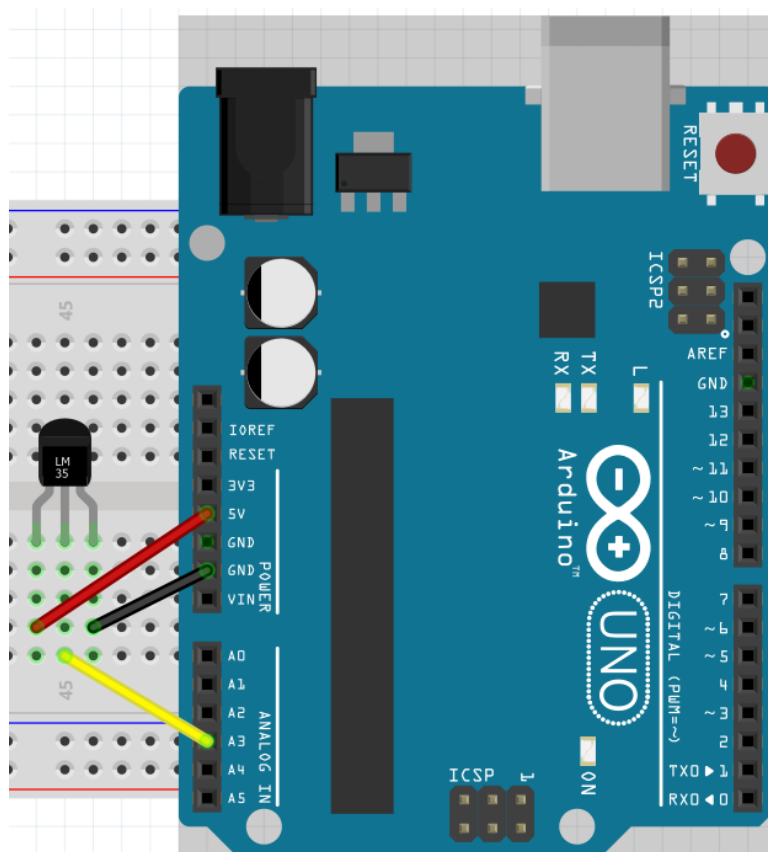
תחום הטמפרטורות שאפשר למדוד באמצעות החיישן הוא, בקירוב, מ- 50°C עד 150°C .



איור 3.9 הדקי חיישן הטמפרטורה מדגם LM 35

באיור 3.9 תיאור סכמטי של הרכיב LM35 והדקיו.

דוגמה 3.4 – LM35 לכניסה אנלוגית A3



איור 3.10 דיאגרמת חיבורים של LM35 למיקרו-בקר

כתבו תוכנית להצגת הטמפרטורה בחלון ה-Serial Monitor, שתציג את עוצמת המתח הנמדד ואת הטמפרטורה.

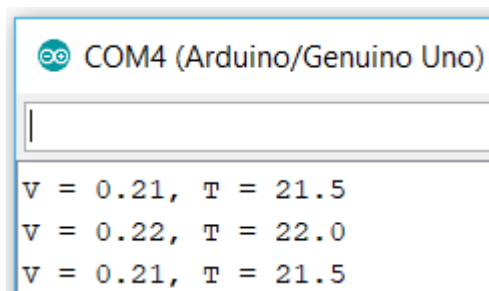
הנוסחה לחישוב הטמפרטורה:

$$T = \frac{V}{10mV} \quad [^{\circ}C]$$

```
int analogPin = A3 ;
int val = 0; // ערך ספרתי
float voltage; // ערך מתח מחושב
float temp; // ערך טמפרטורה מחושב
void setup () {
    Serial.begin(9600); // תיחול תקשורת טורית
}

void loop () {
    val = analogRead(analogPin) ; // קריאת ערך אנלוגי
    voltage = 0.00488 * val; // חישוב ערך המתח
    temp = voltage / 0.01; // חישוב ערך הטמפרטורה
    Serial.print("V = ");
    Serial.print(voltage);
    Serial.print(", T = ");
    Serial.println(temp,1) ; // הצגת הטמפרטורה ספרה אחת אחרי הנקודה
}
```

דוגמת פלט:



```
COM4 (Arduino/Genuino Uno)
V = 0.21, T = 21.5
V = 0.22, T = 22.0
V = 0.21, T = 21.5
```

תרגול 

שאלה 3.8

חיישן טמפרטורה מחובר כמו באיור 3.10. מחברים 2 נורות לד ליציאות ספרתיות 6 ו-7. הנוריות המחוברות בתצורת pull down. אם הטמפרטורה נמוכה מ- 25°C , תדלוק נורת הלבד המחוברת להדק 6. אם הטמפרטורה גבוהה מ- 35°C , תדלוק נורת הלבד בהדק 7. בכל מקרה אחר נורות הלבד יהיו כבויות. כתבו תכנית להפעלת הנוריות כנדרש.

שאלה 3.9

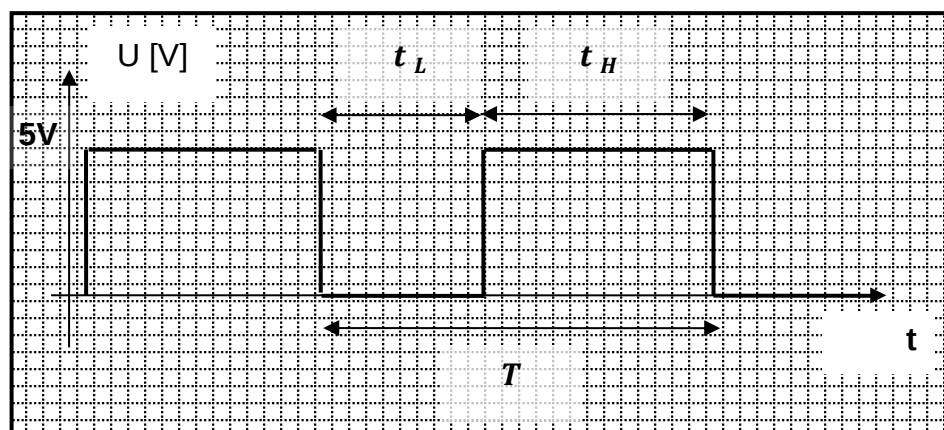
חיישן טמפרטורה מחובר כמו באיור 3.10. מחברים נורת לד ליציאה הספרתית 6 וזמזם ליציאה הספרתית 7. קיבעו את תצורת החיבור של הנורה והזמזם כרצונכם. אם הטמפרטורה גבוהה מ-55°C, יופעל הזמזם. אם הטמפרטורה תהיה מ-10°C עד 25°C, נורת הלד תדלוק. כתבו תכנית להפעלת הנורה והזמזם כנדרש.

3.5 מוצא אנלוגי

כאמור, מכיוון שבפועל אין מוצא אנלוגי, משתמשים ביציאה ספרתית הפועלת על פי העיקרון של אפנון רוחב הדופק PWM (Pulse Width Modulation).

מאפייני גל ריבועי

- עוצמת הגל – מתח 0V או 5V
- תדירות הגל $f [Hz]$
- זמן המחזור $T [sec]$
- גורם המחזור DC (Duty Cycle) (מקובל לציין באחוזים).



איור 3.11 גל ריבועי

גורם המחזור הוא היחס בין פרק הזמן שבו עוצמת הגל הריבועי 5V לבין המחזור:

$$DC = \frac{t_H}{T}$$

$$T = t_H + t_L \quad \text{נשים לב כי}$$

שיטת PWM מבוססת על העיקרון שלפיו התדר קבוע אולם גורם המחזור משתנה לפי הצורך. עקב שינוי גורם המחזור משתנה המתח הממוצע של האות.

המתח הממוצע הוא המתח הישר השקול לגל הריבועי, וזו הנוסחה לחישובו:

$$V_{AV} = \frac{5V \times t_H}{T} = 5V \times \frac{t_H}{T} = 5V \times DC$$

על פי הנוסחה המתח הממוצע תלוי בגורם המחזור DC בלבד!

ערכו של גורם המחזור הוא בתחום: $0\% \leq DC \leq 100\%$. מכאן שהמתח הממוצע יהיה בתחום $0 \leq V_{AV} \leq 5V$.

מסקנה: בממוצע אפשר להפיק אות מתח בתחום שמ-0V עד 5V, בתלות בגורם המחזור.

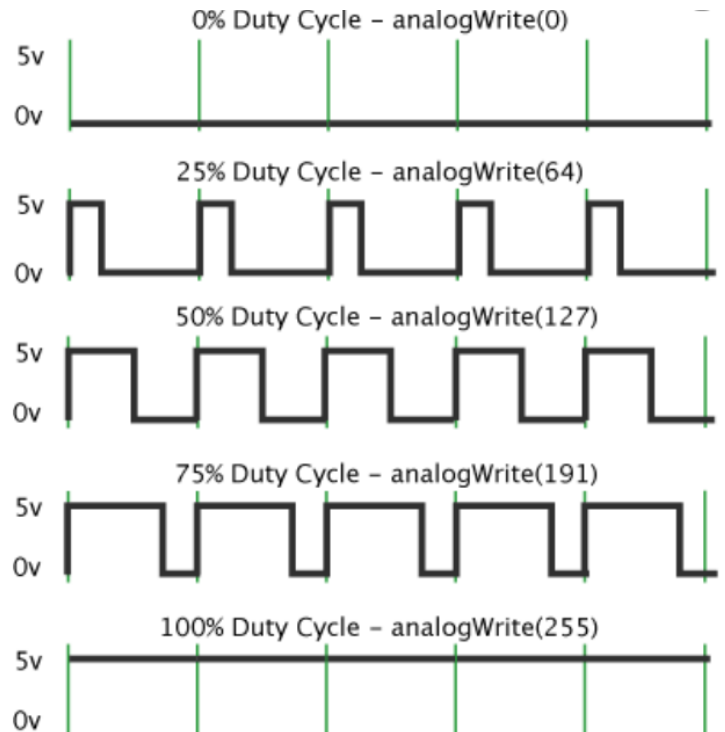
כדי לייצר אות PWM בתוכנה נשתמש במספר המייצג את גורם המחזור. המספר הוא בתחום שמ-0 עד 255. ההתאמה בין גורם המחזור, המתחים והערך המספרי להלן:

$$0\% \leq DC \leq 100\%$$

$$0 \leq V_{AV} \leq 5V$$

$$0 \leq Value \leq 255$$

באיור 3.12 מתואר הקשר בין הערך המספרי הדרוש לכתיבה לערוץ האנלוגי ובין גורם המחזור שמתקבל.



איור 3.12 הקשר בין הערך המספרי ובין גורם המחזור

כתיבת הערך האנלוגי

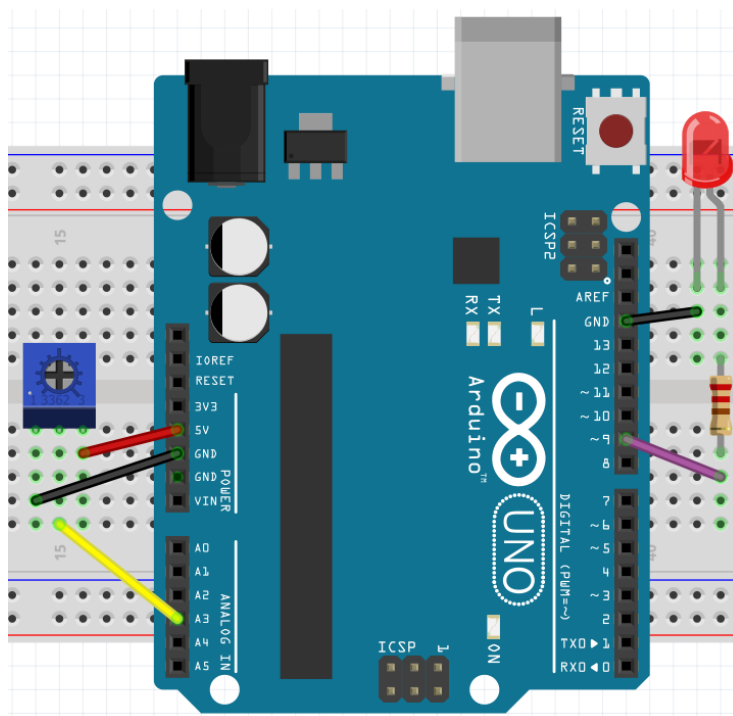
ההוראה `analogWrite(pin, value)` משמשת לכתיבת הערך האנלוגי להדק המתאים. הוראה זו גורמת ליצירת האות הריבועי בהדק המתאים עד להוצאת ערך אנלוגי חדש לאותו הדק או קריאה או כתיבה של ערך ספרתי להדק זה.

למשל, ההוראה `analogWrite(3, 127);` תגרום ליצירת גל ריבועי בגורם מחזור של 50%
בהדק 3.

אפשר לייצר גל ריבועי בשישה הדקים המסומנים בסימן "~" על לוח UNO. מספרי ההדקים הם: 3,5,6,9,10,11. בהדקים 3,9,10,11 נוצר אות בתדר 490Hz בקירוב ואילו בהדקים 5 ו-6 נוצר אות בתדר 980Hz בקירוב (בגלל שימוש נוסף שנעשה ביחידות הקשורות להדקים האלה).

דוגמה 3.5 – הדלקת נורת לד בעוצמה התלויה בכניסה אנלוגית

כתוב תוכנית המפעילה את דיודת הילד המחוברת להדק 9 כאשר לכניסה אנלוגית 0 מחובר פוטנציומטר.



איור 3.13 מעגל בקרת העוצמה של הארת דיודת לד באמצעות פוטנציומטר


```

int pinLed = 9;           // הגדרת הדק הלד
int analogPin = 3;       // הגדרת הדק הכניסה האנלוגית
int val;                 // הגדרת משתנה לקליטת הערך האנלוגי

void setup() {
    pinMode(pinLed, OUTPUT); // הגדרת הדק הלד כמוצא
}

void loop() {
    val = analogRead(analogPin); // קריאת הכניסה האנלוגית
    analogWrite(pinLed, val / 4); // כתיבת ערך האנלוגי לדיודת הלד
}

```

הסבר

הערך שנקרא מהכניסה האנלוגית הוא בתחום שמ-0 עד 1023 ואילו הערך שאפשר לכתוב להדק מוצא אנלוגי הוא בתחום שמ-0 עד 255. כדי להתאים את הערכים נחלק את הערך של val בארבע, וכך נקבל את ערך התחום המתאים.



שאלה 3.10

כתבו תוכנית המפעילה נורת לד המחוברת ליציאה אנלוגית 3 כך שעוצמת ההארה תתחיל מערך מינימלי (כבוי), תעלה בהדרגה כל 1mSec עד לערך מרבי ולאחר מכן תקטן עוצמתה בהדרגה כל 1mSec עד לעוצמה מינימלית. התהליך יחזור בלולאה אין-סופית.

שאלה 3.11

כתבו תוכנית שתפעיל נורת לד שמחוברת ליציאה אנלוגית 11 כך שעוצמתו תתחיל מערך מינימלי (כבוי), תעלה בהדרגה כל 2mSec עד לערך מרבי ולאחר מכן הנורה תהיה כבויה למשך חצי שנייה. התהליך יחזור בלולאה אין-סופית.

יצירת גל ריבועי

ההוראה **tone()** משמשת ליצירת גל ריבועי בתדר רצוי. לגל הריבועי גורם מחזור של 50%. אפשר ליצור גל ריבועי בכל אחד מהדקים. יש אפשרות ליצור גל ריבועי למשך זמן קצוב או עד לשינוי התדירות. הפסקת הגל הריבועי נעשית באמצעות הפונקציה **noTone()**.

תחביר הפונקציה **tone**

```
tone(pin, frequency);
tone(pin, frequency, duration);
```

pin – מספר הדק

frequency – מספר שלם חיובי

duration – משך הזמן שהגל הריבועי יופק בו, במילי-שניות (מספר שלם ארוך).

הערות

בזמן נתון אפשר להפיק גל ריבועי אחד. אם הגל הריבועי כבר מיוצר, לא תתאפשר יצירת גל ריבועי בהדק אחר. אם נפנה לאותו הדק כאשר כבר נוצר גל ריבועי, ישתנה התדר של הגל הריבועי.

בזמן יצירת הגל הריבועי יופרעו אותות PWM בהדקים 3 ו-11.

התדר הנמוך ביותר שאפשר לייצר הוא 31Hz.

למשל, ההוראה **tone(3, 1000, 1000)** תגרום ליצירת גל ריבועי בתדר 1kHz עם גורם מחזור של 50% בהדק 3 למשך שנייה אחת.

אם רוצים לייצר גל ריבועי בהדקים שונים, חייבים להפסיק באמצעות הפונקציה **noTone** את הגל הריבועי בהדק הקיים ולהפעיל אותו בהדק אחר.

תחביר הפונקציה **NoTone**

```
noTone(pin)
```

pin – מספר הדק

הערות

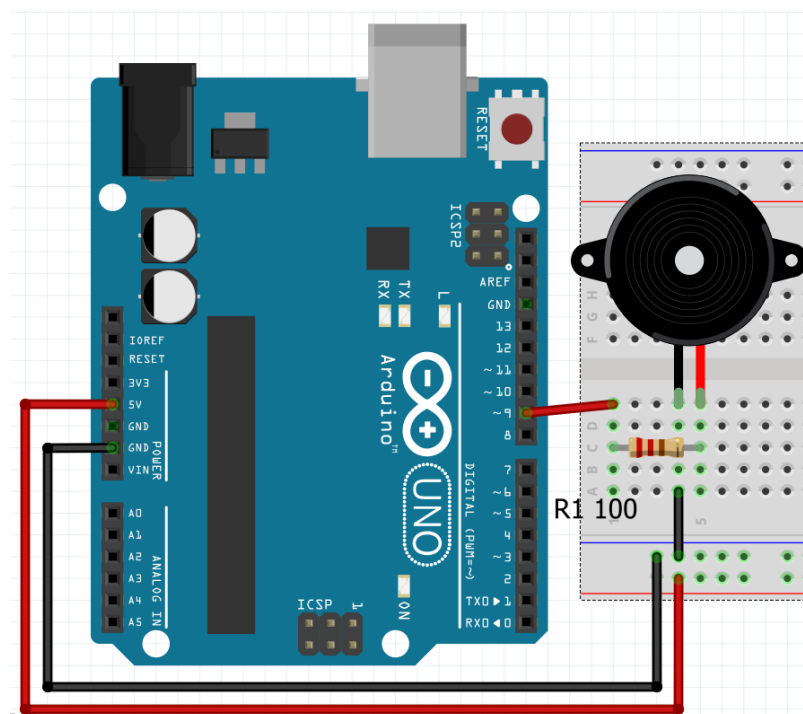
הפעולה מפסיקה את הגל הריבועי בהדק המתאים.

אם לא נוצר גל ריבועי בהדק, אין לפונקציה השפעה.

דוגמה 3.6 – יצירת צליל מופסק

כתבו תוכנית ליצירת צליל בתדר 500Hz למשך שנייה אחת עם הפסקה של שתי שניות.

הצליל יופק ברמקול המחובר להדק 9 במיקרו-בקור.



איור 3.13 מעגל הפעלת רמקול

```
int speakerPin = 9;
void setup() {
    pinMode(speakerPin, OUTPUT);
}
void loop() {
    tone(speakerPin, 500, 1000);
    delay(2000);
}
```

שאלה 3.12

מעוניינים ליצור נגן עם ארבעה לחצנים כך שכל לחצן ישמיע צליל בתדר שונה.

מספר לחצן	תדירות הצליל
1	262
2	294
3	330
4	349

כל אחד מהצלילים יושמע למשך 0.25Sec.

- א. שרטטו מעגל חשמלי לחיבור ארבעת הלחצנים והרמקול למיקרו-בקר.
- ב. כתבו תוכנית להשמעת צליל בכל לחיצה על לחצן מסוים.
- ג. חברו את המעגל החשמלי ללוח UNO ובדקו את התוכנית שכתבתם. הוסיפו לתוכנית הוראות להצגת מצב התוכנית בשימוש בהוראות Serial.print או Serial.println.

שאלה 3.13

סולם הצלילים מורכב משבעה תווים. לכל תו תדר אחר.

תו	דו	רה	מי	פה	סול	לה	סי
תדר	262	294	330	349	392	440	495
סמל	C	D	E	F	G	A	B

נתון קטע הנגינה הבא (משמאל לימין):

C C C G A A
E E D D C

כתבו תוכנית המנגנת קטע נגינה זה. כל תו בשורה הראשונה ינוגן במשך רבע שנייה. הרווחים בין התווים הם הפסקות באורך רבע שנייה. בשורה השנייה התווים ינוגנו במשך רבע שנייה ללא הפסקה.

נסו לזהות את המנגינה.

סיכום פרק 3

בפרק 3 למדתם את הנושאים האלה:

1. הגדרת אות אנלוגי
2. ההבדל בין אות אנלוגי לאות ספרתי
3. המרת אות אנלוגי לספרתי
4. כושר הבחנה – רזולוציה
5. מבואות אנלוגיים A0, A1, A2, A3, A4, A5
6. פונקציית המרה `analogRead`
7. הזמן הדרוש להמרה והשהיה בין קריאות `delayMicroseconds`
8. חיבור פוטנציומטר למבוא אנלוגי
9. חיבור חיישן אור התנגדותי LDR למבוא אנלוגי
10. חיבור חיישן טמפרטורה LM35 למבוא אנלוגי
11. מוצאים אנלוגיים
12. מאפייני גל ריבועי: תדר, זמן מחזור וגורם מחזור
13. שיטת PWM ומתח ממוצע
14. מוצאים אנלוגיים בהדקים 3,5,6,9,10,11
15. פונקציית המרה `analogWrite`
16. יצירת גל ריבועי באמצעות פונקציית `tone`
17. עצירת גל ריבועי באמצעות פונקציה `noTone`
18. יצירת צלילים ברמקול

פרק 4: מיקרו-בקרים

בפרק הראשון הכרנו את המונח מיקרו-בקר (MCU – Micro Controller Unit). התייחסנו אליו כאל מחשב זעיר הכולל את שלוש היחידות הבסיסיות של כל מערכת מחשב: מעבד, זיכרון והתקני קלט/פלט.

4.1 מיקרו-בקר

בקר (Controller) הוא מחשב שנועד לבקר תהליכים או פעולה של התקן כלשהו. בקר אחראי, למשל, לבקרת הטמפרטורה במערכת מיזוג אוויר.

מיקרו-בקר הוא בקר הדחוס לתוך שבב יחיד.

תכונות המיקרו-בקר

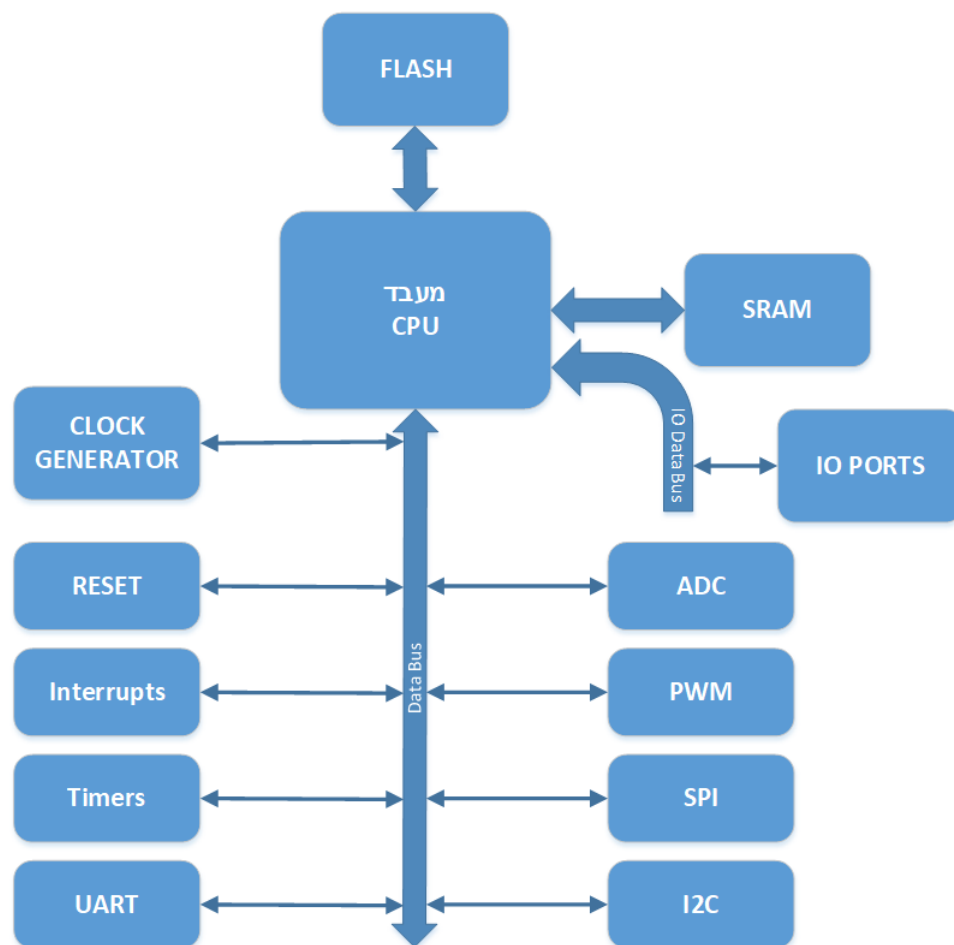
- בקר מטפל בדרך כלל במשימה אחת שמספר הנתונים בה קטן.
- מרחב הזיכרון שלו אינו גדול.
- מהירות הפעולה שלו אינה גבוהה.
- לבקר רכיבי עזר מובנים המסייעים בפעולתו.

רכיבי העזר בבקר

- **קוצבי זמן** – משמשים לקציבת זמן ולמדידת זמן.
- **מונים** – משמשים למניית אירועים.
- **בקרי תקשורת טורית** – משמשים ליצירת תקשורת בין שני בקרים או בין בקר להתקנים אחרים.
- **בקר פסיקות** – משמש לניהול בקשות שירות של מעגלים פנימיים וחיצוניים המתריעים על אירועים שונים.
- **אפנון רוחב הדופק PWM** – משמש ליצירת דופק בעל רוחב משתנה.
- **ממירים A/D ו-D/A** – משמשים לקריאה וכתיבה של אותות אנלוגיים.
- **משוויס אנלוגיים** – משמשים לזיהוי תחום של ערכים אנלוגיים.

באיור 4.1 נראות היחידות הבסיסיות של מיקרו-בקר. לב המערכת הוא **המעבד**, המחובר לשני מרחבי **זיכרון** עיקריים: זיכרון הבזק (Flash), המשמש בדרך כלל לאחסון התוכנית וערכים קבועים, וזיכרון גישה אקראית סטטי (SRAM – Static RAM), המשמש לאחסון נתונים. במיקרו-בקרים רבים יש גם זיכרון מסוג EEPROM, שאינו נראה באיור, והוא משמש להפעלה ראשונית של המיקרו-בקר (boot) בשלב הכתיבה (upload) של תוכנית.

פס נתוני קלט/פלט (IO Data Bus) המחבר את הדקי המיקרו-בקר – מפתחי קלט/פלט – למעבד. לכמה מההדקים שימושים נוספים על אלה של הדקי IO.



איור 4.1 מבנה מיקרו-בקר

Clock Generator – מחולל אות שעון עבור המיקרו-בקר. אפשר לקבוע את קצב העבודה של מיקרו-בקר על ידי חיבור גביש חיצוני, ובמיקרו-בקרים מסוימים אפשר לקבוע את קצב העבודה באמצעות שעון פנימי מובנה.

Reset – מערכת המאתחלת את המיקרו-בקר. מקורות האיפוס מגוונים, ובהם כניסה מיוחדת שבדרך כלל מחוברת למפסק לחצן, מנגנון הנקרא watch dog ואיפוס עקב נפילת מתח.

ADC – מערכת הממירה אותות אנלוגיים לאותו ספרתיים.

PWM – מערכת המייצרת אותות ריבועיים ברוחב משתנה.

Interrupts – מנגנון לניהול בקשות טיפול באירועים חיצוניים (הדק המחובר למעגל חשמלי) או באירועים פנימיים (קוצבי זמן, מונים, תקשורת טורית).

Timers – מונים המשמשים לספירת אירועים או לקציבת זמן.

UART – מערכת המקשרת בין שני מיקרו-בקרים או בין מיקרו-בקר להתקן אחר ואחראית לשידור וקליטה של מידע בתקשורת טורית.

I2C – מערכת לחיבור התקנים למיקרו-בקר במרחקים קצרים בתקשורת טורית. התקשורת היא בשיטה דו-כיוונית למחצה (Half Duplex).

SPI – מערכת לחיבור התקנים למיקרו-בקר במרחקים קצרים בתקשורת טורית. התקשורת היא בשיטה דו-כיוונית מלאה (Full Duplex).

בפרקים הקודמים למדנו כיצד להשתמש בכניסות ויציאות ספרתיות, בכניסה אנלוגית המתחברת אל ממיר ADC וביציאה אנלוגית שנעשה בה שימוש בהדק ספרתי ליצירת אותות PWM.

בפרק הבא נלמד כיצד להשתמש בכניסות ספרתיות כדי לקבל מידע ממעגלים חיצוניים על התרחשות אירוע וכיצד להגיב על אירוע כזה ועל אירוע שמקורו בקוצבי זמן.

בפרק האחרון נלמד כיצד להשתמש בתקשורת הטורית UART לחיבור מחשב אל לוח UNO. השימוש בתקני I2C ו-SPI חורג מגבולות תוכנית הלימודים בשלב זה ולכן לא נתייחס אליו כאן.

4.2 ארכיטקטורת RISC

ארכיטקטורת מחשב היא קבוצת חוקים ושיטות להגדרת התפקוד, הארגון והיישום של מערכות מחשב. אחד ההיבטים של ארכיטקטורת מחשב היא **ארכיטקטורת קבוצת ההוראות (ISA – Instruction Set Architecture)**.

ארכיטקטורת קבוצת ההוראות היא הממשק בין תוכנת המחשב ובין החומרה. אפשר להתייחס לארכיטקטורה זו כאל נקודת המבט של המתכנת על המכונה (כלומר המעבד).

המעבד אינו "מבין" שפות עיליות כגון Java, C# או C++. הוא מסוגל להבין רק הוראות המקודדות בצורה מספרית, בדרך כלל במספרים בינאריים. תוכנות כגון מהדרים (Compilers) מתרגמות את השפה העילית לשפה הברורה למעבד (שפת מכונה).

פרט להוראות, הארכיטקטורה מגדירה פריטים נוספים הזמינים לתוכנית: טיפוס נתונים, אוגרים, שיטות מעון וזיכרון. ההוראות חלות על הפריטים האלה באמצעות מספרי אוגרים (שמות) ושיטות מעון לזיכרון.

אפשר לסווג את הארכיטקטורה בכמה דרכים; סוג הארכיטקטורה הנפוץ הוא ארכיטקטורה מורכבת המוכרת בשם CISC (Complex Instruction Set Computer). בארכיטקטורה זו יש הוראות רבות המתמחות בפעולות מסוימות, שהשימוש בחלק מהן עשוי להיות נדיר. הארכיטקטורה RISC (Reduced Instruction Set Computer), הוראות מחשב מופחתות, מפשטת את המעבד ביישום יעיל של הוראות המשמשות תוכניות באופן תדיר, ואילו הוראות שנדרשות רק לעיתים רחוקות מיושמות באמצעות תת-תוכניות (פעולות). אומנם השימוש בתת-תוכניות מאריך את משך הביצוע, אבל הפיצוי על כך הוא בתדירות הנמוכה שבה משתמשים בהוראות האלה.

בשיטת CISC מספר הבתים הדרוש לכתיבת כל הוראה וזמן הביצוע שלה אינם אחידים אלא תלויים במורכבות ההוראה, ואילו בשיטת RISC לכל הוראה מספר בתים אחיד ומשך הביצוע של כל הוראה הוא מחזור אחד בלבד.

בארכיטקטורת CISC אפשר לפנות לזיכרון לשם קריאה בהוראה אחת, לאחר מכן לבצע את הפעולה ולבסוף לכתוב את הנתונים בזיכרון, ואילו ב-RISC יש הפרדה ברורה בין הפעולות: פנייה לזיכרון נעשית בהוראה אחת בלבד ומילוי הוראת פעולה נעשה במעבד באמצעות אוגרים בלבד. משום כך נדרשים בארכיטקטורת RISC אוגרים רבים וזיכרון תוכנית גדול יותר.

דוגמה 4.1: השוואת מילוי הוראות בשתי שיטות

נדגים פעולת חיבור בין שני תאי זיכרון במעבדים דמיוניים בשתי השיטות.

הפעולה בשפה עילית תיראה כך: $a = a + b$

נניח כי למשתנה a כתובת 100 ולמשתנה b כתובת 200.

נניח כי בשני המעבדים פעולת החיבור נקראת ADD.

ההוראה במעבד CISC תיראה כך:

ADD 100, 200

כלומר: חבר את הערך שבתא 100 עם הערך שבתא 200 ואחסן בתא 100.

ההוראות במעבד RISC ייראו כך:

LOAD Reg1, 100

LOAD Reg2, 200

ADD Reg1, Reg2

STORE 100, Reg1

נשים לב לכך שכאן מתמלאות ארבע הוראות במקום אחת:

- ההוראה LOAD טוענת את הנתון מתא זיכרון 100 אל האוגר Reg1.
- ההוראה LOAD טוענת את הנתון מתא זיכרון 200 אל האוגר Reg2.
- פעולת החיבור ADD נעשית במעבד באמצעות האוגרים

$$\text{Reg1} = \text{Reg1} + \text{Reg2}$$

- ההוראה STORE מאחסנת את התוצאה שב-Reg1 בתא זיכרון 100.

בשיטת הפעולה של CISC, כאשר יש הוראות הסתעפות (דילוג), ויש לדלג למקום אחר בתכני תוכנית, נדרש זמן רב כדי לקבל את ההוראה החדשה לביצוע, ואילו בארכיטקטורת RISC יש מנגנון לחיזוי ההוראה הבאה לביצוע.

בחירת הארכיטקטורה משפיעה במידה רבה על תכנון המעבד והזיכרון. בשיטת CISC על המעבד לפענח מגוון רחב של הוראות ועל כן הוא חייב להיות מורכב מהמעבד בארכיטקטורת RISC, משום שמספר ההוראות לפענוח בשיטת RISC מצומצם. מצד שני, הדגש על פנייה לזיכרון גדול יותר בשיטת RISC, שכן נדרשת פנייה מרובה לזיכרון בשל מספר ההוראות שיש לבצע.

4.3 מבנה הפנימי של מיקרו-בקר ATmega328P

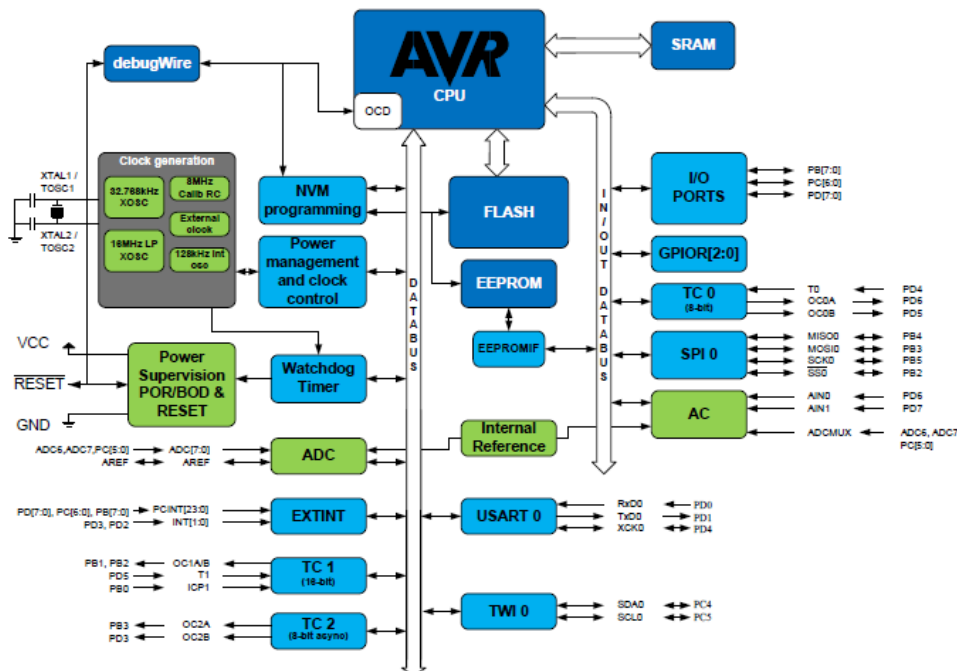
המיקרו-בקר ATmega328P של היצרנית Atmel® מבוסס על ארכיטקטורת AVR® enhanced RISC ומיוצר בטכנולוגיית CMOS עם צריכת הספק נמוכה. מיקרו-בקר זה פועל עם מידע ברוחב 8 סיביות (8-bit). ההוראות בו מתבצעות במחזור פעולה יחיד, כלומר בקצב של 16MHz יתבצעו כ-16 מיליון פעולות בשנייה.

תכונות

- ארכיטקטורת RISC 8 סיביות
 - 131 הוראות
 - רוב ההוראות מתבצעות במחזור יחיד
 - 32 אוגרים לשימוש כללי
- מרחבי זיכרון לא נדיפים
 - זיכרון Flash בגודל 32KB
 - זיכרון SRAM בגודל 2KB
 - זיכרון EEPROM בגודל 1KB
- רכיבים היקפיים
- שני מונים/ קוצבי זמן של 8 סיביות
- מונה/קוצב זמן אחד של 16 סיביות
- מונה זמן אמת עם מתנד נפרד
- שישה ערוצי PWM

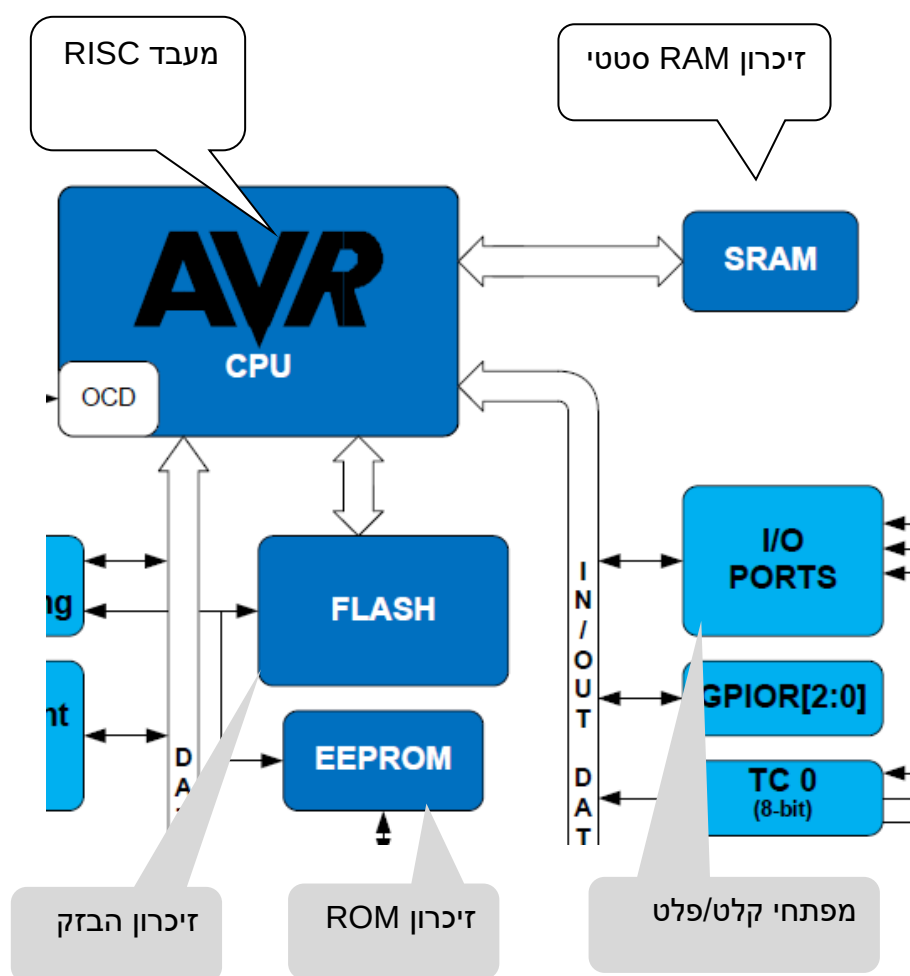
- שישה ערוצי המרה מאנלוגי לדיגיטלי ברוחב 10 סיביות
- שני ממשקים טוריים SPI
- תקשורת טורית אחת USART
- קוצב זמן Watchdog עם מתנד מקומי נפרד
- משווה אנלוגי
- מאפיינים מיוחדים
 - איפוס בזמן הפעלה
 - מתנד פנימי
 - מקורות פסיקה פנימיים וחיצוניים
- קלט/פלט I/O
 - 23 קווי I/O בני תכנות
- מתחי פעולה
 - 1.8-5.5V

באיור 4.2 נראית דיאגרמת מלבנים של המיקרו-בקר.



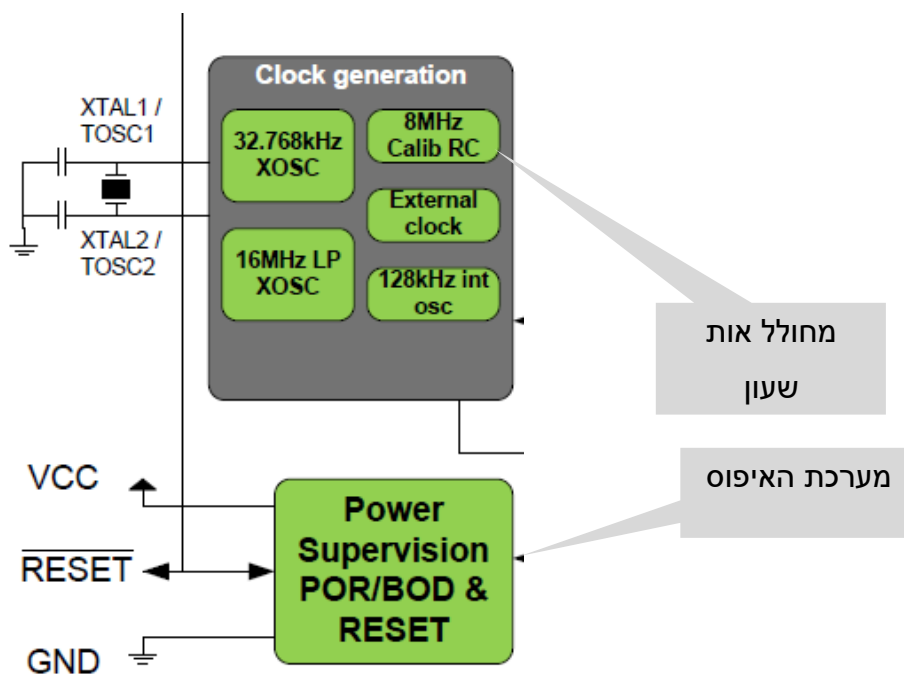
איור 4.2 דיאגרמת מלבנים של המיקרו-בקר ATmega328P

באיור 4.3 נראים מרכיבי המחשב היסודיים: מעבד, זיכרון וקלט/פלט.



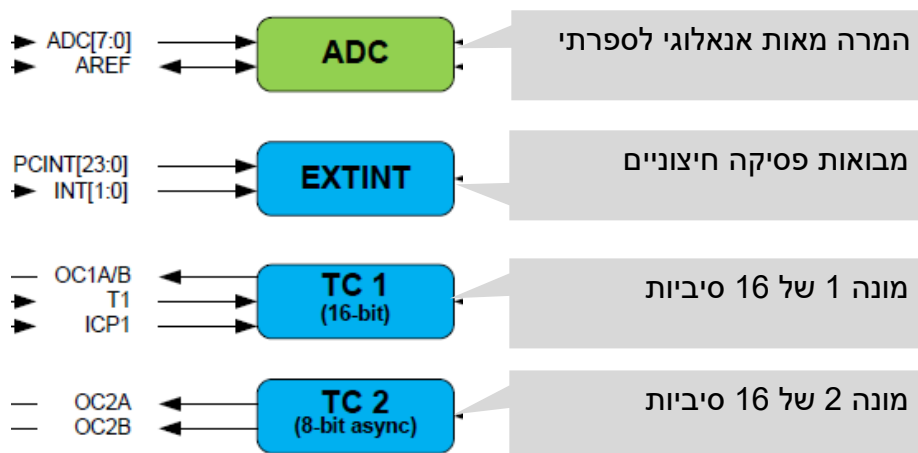
איור 4.3 המעבד, יחידות הזיכרון ומפתחי קלט/פלט

באיור 4.4 נראות מערכת מחולל אות השעון ומערכת האיפוס.



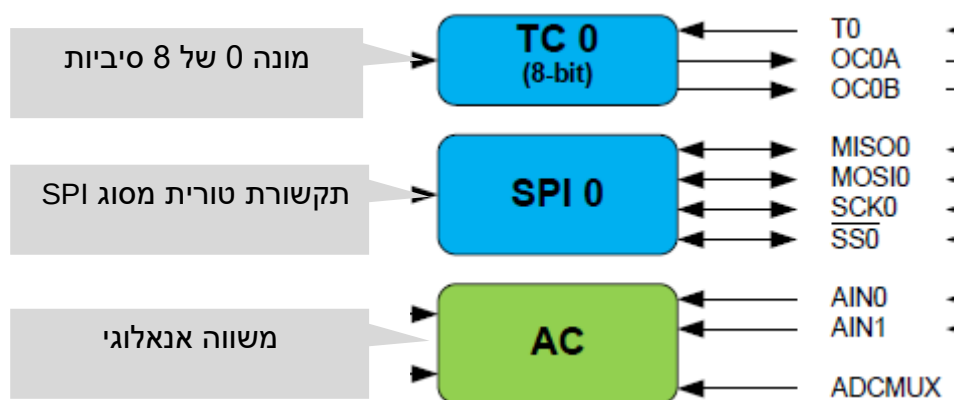
איור 4.4 מחולל אות השעון והאיפוס

איור 4.5 מתאר את רכיבי העזר במיקרו בקר



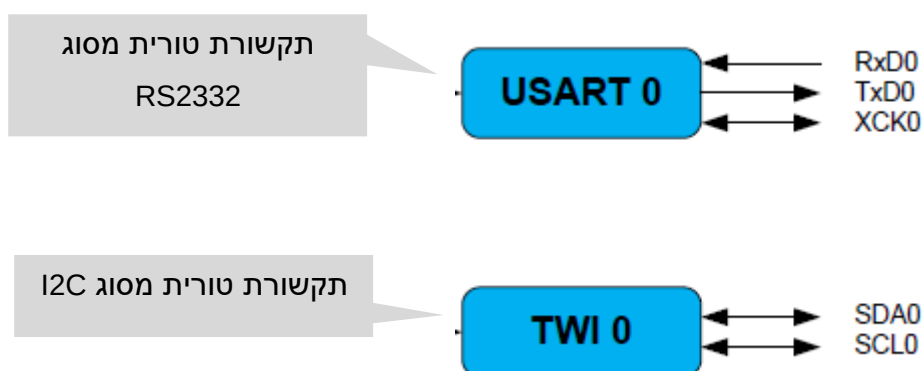
איור 4.5 רכיבי עזר: פסיקה חיצונית, ממיר ADC ומוני 16 סיביות משווה אנאלוגי

באיור 4.6 נראית קבוצה נוספת של רכיבי עזר



איור 4.6 מונה 8 סיביות, משוואה אנלוגי ותקשורת טורית SPI

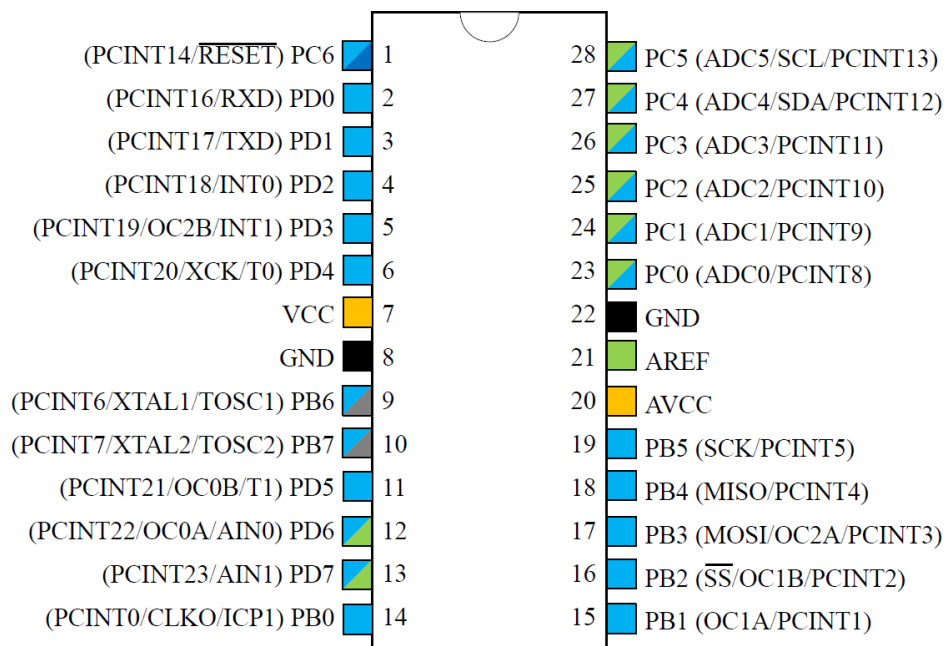
באיור 4.7 נראית קבוצה אחרונה של רכיבי עזר



איור 4.7 בקר תקשורת טורית RS232 ותקשורת טורית I2C

4.4 מבנה האריזה של המיקרו-בקר

באיור 4.8 מוצגים ההדקים של המיקרו-בקר.



איור 4.8 הדקי המיקרו-בקר ATmega328P

ביאור האיור:

הדקים עם תפקיד יחיד

הדק 7 : VCC מתח אספקה 5V למערכת הספרתית

הדק 8 והדק 22 : GND אדמה

הדק 20 : AVCC מתח אספקה לממיר האנלוגי לספרתי. הדק זה יחובר למקור 5V גם אם לא משתמשים בממיר. המתח יחובר דרך מסנן

הדק 21 : AREF מתח ייחוס למערכת ההמרה מתקבילי לספרתי

הדקים מרובי תפקידים

ליד כל הדק מופיעים שם המפתח שאליו הוא משויך, מקומו במפתח ותפקידיו הנוספים.

הדקים 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

הדקים 3, 4, 5, 6, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

הדקי גביש

כדי לקבוע את קצב העבודה של המיקרו-בקר מחברים אליו חיצונית רכיב הנקרא גביש. הגביש מחובר להדקים PB6/XTAL1 ו-PB7/XTAL2 ובעצם אין אפשרות להשתמש בשני הקווים האלה למטרות אחרות.

הדק איפוס

כדי לאתחל (לאפס) את המיקרו-בקר מחברים מפסק להדק ה-PC6/RESET. לחיצה על המפסק גורמת למיקרו-בקר להפעיל תוכנית ראשונית שנמצאת במרחב זיכרון התוכנית בכתובת 0.

הדקים ספרתיים

ההדקים מקבוצת PB[0..5], PD[0..7], השייכת למפתחים B ו-D, שאינם בשימוש, משמשים כקווים ספרתיים.

הדקי מבוא אנלוגיים

ההדקים מקבוצת PC[0..5], השייכת למפתח C, יכולים לשמש הדקים ספרתיים או הדקים אנא לאנלוגיים.

השימוש בהדקי הגביש ובהדק האיפוס מצמצם את מספר ההדקים העומדים לרשותנו, וכך בעצם יש לנו 14 קווי I/O ספרתיים ושישה קווים אנלוגיים (אפשר להשתמש בקווים האנלוגיים גם כקווים ספרתיים).

תקשורת טורית UART

ההדקים PD0/RXD ו-PD1/TXD משמשים כקווי תקשורת לקליטה ולשידור, בהתאמה. בלוח UNO נעשה שימוש רב בהדקים האלה, ולכן על פי רוב לא נוכל להשתמש בהם.

מבואות פסיקה

ההדקים PD2/INT0 ו-PD3/INT1 משמשים, לפי הצורך, כמבואות פסיקה חיצוניים.

רכיבי עזר נוספים

יש עוד שלושה רכיבי עזר: תקשורת טורית SPI, תקשורת טורית I2C ומשווה אנלוגי. ברכיבי העזר האלה נעסוק במסגרת ההרחבה הנלמדת במקצוע ההתמחות.

4.5 המפרט הטכני של לוח Arduino UNO

הלוח מבוסס על המיקרו-בקר ATmega328P. ללוח 14 הדקים המשמשים לכניסה או ליציאה ספרתית, שש כניסות תקביליות, מתנד גבישי בתדר 16MHz, חיבור USB, שקע אספקת מתח, לחצן איפוס וממשק חיבור ICSP.

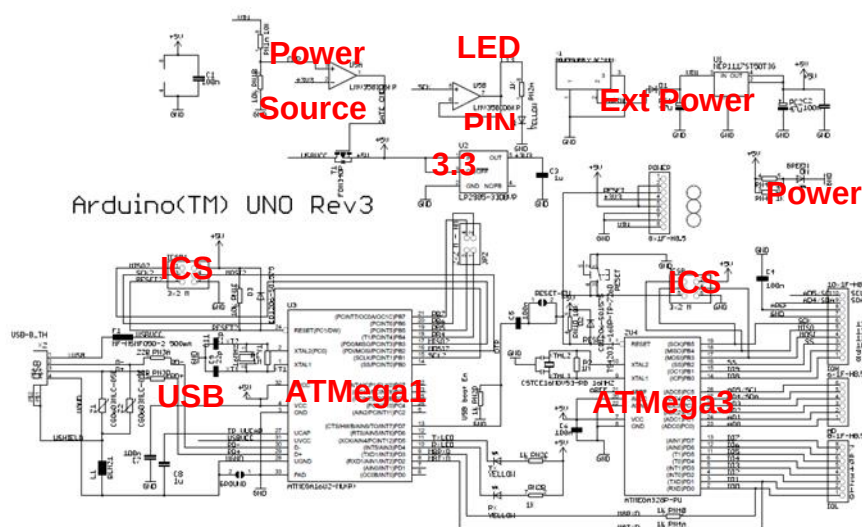
נוסף על אלה יש בלוח ארבע נורות לד: נורת חיווי פעולה ON, נורת לד לשימוש כללי המחוברת להדק 13, ושתי נורות לד לחיווי פעולה של תקשורת בין המיקרו-בקר למחשב.

מפרט טכני

מיקרו-בקר	ATmega328P
מתח פעולה	5V
מתח מבוא בשקע המתח (מומלץ)	7-12V
הדקי קלט/פלט ספרתיים (דיגיטליים)	14 (שישה מהם משמשים כיציאות I/O)
הדקי I/O PWM	שישה מ-14 הדקי I/O ומסומנים ב-~ 3, 5, 6, 9, 10 ו-11
הדקי I/O תקשורת	RX – 0, TX – 1 (מתוך 14 הדקי I/O)
כניסות תקביליות (אנלוגיות)	שש, מסומנות A0 עד A5
זרם לכל הדק I/O	20mA
זרם להדק 3.3V	50mA
תדירות שעון	16MHz
נורת לד מובנית LED_BUILTIN	הדק 13 בלוח
זיכרון FLASH של תוכן התוכנית	32KB
זיכרון RAM סטטי לנתונים	2KB

4.6 התרשים החשמלי של כרטיס הפיתוח Arduino Uno

באיור 4.9 נראה המעגל החשמלי של לוח UNO בגרסה 3.



איור 4.9 תרשים חשמלי של UNO גרסה 3

מיקרו-בקרים

בתרשים החשמלי שני מיקרו-בקרים:

- מיקרו-בקר ATmega16U2 המשמש לתקשורת USB בין המחשב ובין לוח ה-UNO.
- מיקרו-בקר ATmega328P המשמש כמיקרו-בקר בלוח.

לכל אחד משני המיקרו-בקרים יש שקע חיבור ICSP¹ המשמש לתכנות המיקרו-בקר כאשר הוא מחובר בלוח.

למיקרו-בקר ATmega16U2 מעגל חשמלי הכולל את שקע חיבור ה-USB שלו למחשב.

המיקרו-בקרים מחוברים בתקשורת טורית UART, המאפשרת להעלות מהמחשב תוכנית למיקרו-בקר ATmega328P באמצעות המיקרו-בקר ATmega16U2 ולהעביר מידע טורי מהמיקרו-בקר ATmega328P לתוכנית Serial Monitor במחשב באמצעות המיקרו-בקר ATmega16U2.

מיקרו-בקר ATmega16U2 מחובר למבוא האיפוס של המיקרו-בקר ATmega328P, וכך בסיום שלב ההעלאה ניתנת למיקרו-בקר הוראת איפוס והוא יתחיל בביצוע התוכנית החדשה.

¹ ICSP – In Circuit Serial Programming: האפשרות לתכנת מיקרו-בקרים בזמן שהם מחוברים במעגל. בדרך כלל יש לשבבים התומכים בשיטה זו מעגל תמיכה ליצירת מתחי הצריבה הנדרשים.

המיקרו-בקר ATmega16U2 מפעיל שתי נורות לד (TX, RX) לחיווי התקשורת הטורית דרכו.

מתחי הזנה

אפשר להפעיל את המעגל החשמלי משני מקורות:

- שקע חיבור USB המספק מתח של 5V בזרם מרבי של 500mA;
- מחבר מתח Ext Power (ממקור חיצוני) ללוח ה-UNO, המספק מתח המסומן VIN. מתח זה מיוצב, באמצעות מעגל חשמלי מתאים, למתח של 5V. מייצב המתח במעגל מסוגל לספק זרם מרבי של 1A.

אפשר להפעיל את המעגל החשמלי מאחד משני המקורות בלבד – ממקור חיצוני או משקע ה-USB. במעגל החשמלי נכלל מעגל המזהה אם המקור החיצוני מחובר (Power Source Detector). אם מחובר מקור חיצוני, המתח המגיע משקע ה-USB מנותק, והמתח המיוצב המגיע מהמקור החיצוני מספק את המתח למעגל. אם המקור החיצוני אינו מחובר, המתח המתקבל משקע ה-USB מזין את המעגל החשמלי.

במעגל החשמלי נכלל מייצב מתח נוסף, המספק מתח של 3.3V בזרם מרבי של 150mA. נורת לד (Power Led) מחוברת באופן קבוע למתח האספקה כדי לתת חיווי על תקינות המתח.

נורת לד

כל לוחות Arduino מצוידים בנורת לד לשימוש כללי ומחוברת בדרך כלל להדק 13 בלוחות. מעגל ההפעלה של נורת הלד (LED PIN 13) מחובר להדק PB5 במיקרו. ההפעלה נעשית דרך מגבר יחידה (חוצץ) כדי שלא להפריע לאותות המתח בהדק זה כאשר משתמשים בו למטרות אחרות.

תרגול

שאלה 1.1

מה הם הגורמים לכך שבכל מיקרו-בקר יש קוצבי זמן? הביאו דוגמאות לשימוש בקוצבי זמן.

שאלה 1.2

מיקרו-בקרים כוללים שיטות שונות לתקשורת טורית: UART, SPI, I2C, USB.

- מדוע לדעתכם יש שיטות שונות של תקשורת טורית?
- האם התמיכה במגוון רב של שיטות מוצדקת?

שאלה 1.3

מהו AVR? מצאו הסברים באינטרנט. מי הגה את הרעיון?

על השאלות שלהלן השיבו לאחר צפייה בתרשים החשמלי המלא או הורדתו בקישור שלהלן. את התרשים החשמלי אפשר למצוא בחיפוש "arduino uno r3 board" schematic או בקישור:

https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

שאלה 1.4

על פי התרשים החשמלי, האם נורות התקשורת (TX, RX) מושפעות מבחינה חשמלית ישירות מרמות המתח בהדקים PD0 ו-PD1 של המיקרו-בקר ATmega328P?

שאלה 1.5

הסבירו את פעולת המעגל הכולל את מגבר השרת המסומן בתרשים ב-U5A.

שאלה 1.6

מצידו הימני של המיקרו-בקר ATmega328P יש שלושה מחברים.

- א. היכן נמצא כל מחבר בלוח ומה הוא מייצג?
- ב. במחבר המסומן IOH מופיעים הדקים 9 ו-10. מה תפקידם של ההדקים האלה? האם הם מופיעים במחבר נוסף?

שאלה 1.7

מה תפקידו של קבל C1 בתרשים החשמלי?

שאלה 1.8

חפשו באינטרנט הסבר על ICSP. מה תיאור ההדקים המתאים לסדרת ATmega?

סיכום פרק 4

בפרק 4 למדתם את הנושאים האלה:

1. מבנה עקרוני של מיקרו-בקר
2. ארכיטקטורת RISC וארכיטקטורת CISC, יתרונות וחסרונות
3. המבנה הפנימי של המיקרו-בקר ATmega328P
4. תכונות המיקרו-בקר ATmega328P
5. הדקי המיקרו-בקר ATmega328P
6. המפרט הטכני של לוח UNO
7. התרשים החשמלי של לוח הפיתוח
8. התקשורת בין המיקרו-בקרים בלוח
9. מערכת הזנת המתח בלוח

פרק 5: פסיקות חומרה ומונים

בפרקים הקודמים למדנו כיצד לכתוב תוכניות סדרתיות הממלאות הוראות עוקבות זו אחר זו בתוך הפעולה `loop()`.

דוגמה 5.1: הדלקת נורת לד בקצב משתנה לפי מפסק

בדוגמה 8 בפרק 2 כתבנו תוכנית הגורמת להבהוב של דיודת לד המחוברת להדק 13. דרשנו שזמן ההפעלה והכיבוי של נורת הלד יהיה תלוי במצב מפסק המחובר בתצורת Pull-Down להדק 12. כאשר המפסק פתוח נורת הלד תהבהב בקצב של שנייה וכאשר המפסק סגור נורת הלד תהבהב בקצב של חצי שנייה. הפתרון שהוצע בפרק 2 הוא:

```
const int pinLed = 13;           // הגדרת קבוע עבור הדק הלד
const int pinSwitch = 12;        // הגדרת קבוע עבור הדק המפסק
int value;                       // הגדרת משתנה לקליטת מצב המפסק
int timedelay;                   // הגדרת משתנה לקביעת זמן ההפעלה והכיבוי
void setup() {
    pinMode(pinLed, OUTPUT);      // הגדרת הדק הלד כמוצא
    pinMode(pinSwitch, INPUT);    // הגדרת הדק המפסק כמבוא
}
void loop() {
    value = digitalRead(pinSwitch); // קריאת מצב המפסק למשתנה
    if (value == HIGH)
        timedelay = 500;          // מפסק סגור
    else
        timedelay = 1000;         // מפסק פתוח
    digitalWrite(pinLed, HIGH);    // הדלקת דיודת הלד
    delay(timedelay);
    digitalWrite(pinLed, LOW);     // כיבוי דיודת הלד
    delay(timedelay);
}
```

בהתבוננות מחודשת בתוכנית רואים שבתחילת הפעולה `loop()` קוראים את מצב המפסק. לאחר מכן מקבלים החלטות ובעקבותיהן דיודת הלד מופעלת למשך זמן שנקבע על ידי המפסק.

ביצוע התוכנית ישתנה אם נשנה את תנאי הפעלתה, כמפורט להלן:

- מה יקרה אם נאריך את זמן ההדלקה והכיבוי של נורת הLED, למשל לחמש שניות ולעשר שניות במקום לחצי שנייה (500) ולשנייה (1000) בהתאמה?
- מה יקרה אם במקום מפסק נשתמש בלחצן ובמקום לחיצה רצופה כל לחיצה תחליף את זמן ההדלקה והכיבוי?

נראה שהמשתמש בלוח הפיתוח יצטרך ללחוץ לחיצות ארוכות של 10–20 שניות כדי שיהיה אפשר לקרוא את הלחיצה. הסיבה לתופעה זו היא שרוב הזמן תבצע התוכנית את פעולת ההשהיה, וזמן קצר מאוד יוקדש לקריאת מצב המפסק. אם לא נחזיק את המפסק לחוץ לאורך זמן – נחמיץ את רגע הקריאה, והתוכנית תגיב באופן המתאים למצב מפסק לא לחוץ אף שלחצנו על המפסק פעם אחת או יותר.

הפתרון לבעיה זו יהיה להפריד בין קריאת המפסק ובין פעולת ההדלקה והכיבוי.

משמעות ההפרדה כאמור היא שיהיה אפשר **לקרוא** את מצב המפסק **בכל רגע** בתוכני תוכנית, ו**בו בזמן** להמשיך **בפעולת ההדלקה והכיבוי** לפי הזמן שנקבע על ידי המפסק.

לפני שנוכל לממש את רעיון ההפרדה נצטרך לתת את דעתנו על השאלות האלה:

 **האם באמת אפשר לעשות שתי פעולות בו בזמן?**

התשובה היא: לא! במיקרו-בקר זה אפשר לעשות בכל רגע סדרת פעולות אחת בלבד.

 **כיצד אפשר לממש את רעיון ה"בו זמניות"?**

הפתרון הוא להפסיק את התוכנית העיקרית לפרק זמן קצר ככל האפשר, ולהפעיל במקומה תוכנית חלופית שתממש בפרק זמן קצר ככל האפשר. לאחר סיום פעולת התוכנית החלופית חוזרים לתוכנית העיקרית.

 **מה יגרום לעצירת התוכנית העיקרית?**

אפשר לעצור את התוכנית העיקרית רק כאשר מתרחש אירוע המתקבל ממקור חיצוני, כמו מפסק הלחיצה, או מרכיב עזר במיקרו-בקר, כמו קוצב זמן. בשני המקרים מדובר באירוע **שמפריע** לתוכנית העיקרית להתבצע ועוצר אותה זמנית. עצירה זמנית היא בעצם **הפסקה** של התוכנית, כלומר עצירתה ולאחר מכן המשכּתה ממקום העצירה.

5.1 תשאול ופסיקה

בדוגמה 5.1 ראינו תוכנית הפועלת באופן סדרתי; בתחילת פעולת ה-`loop()` התבצעה קריאה של מצב המפסק. בעצם אנו "מתשאלים" את המפסק על מצבו, אם הוא פתוח או סגור. שיטה זו, שבה אנו קוראים את מצב המפסק באופן מחזורי, נקראת שיטת **התשאול** (Polling). תוצאת התשאול קובעת את המשך ביצוע התוכנית.

בדוגמה הצגנו את אחד מחסרונותיה של שיטת התשאול; החיסרון הוא שאנו עלולים להחמיץ אירוע שהתרחש. חיסרון נוסף של שיטת התשאול הוא שהתוכנית תהיה עסוקה מדי בטיפול בתשאול אירועים שונים ולא ייוותר לה זמן לתוכנית העיקרית.

בפתרון שהוצג לדוגמה 5.1 מוצג רעיון השימוש ב**פסיקה** (Interrupt). כאשר מתרחש אירוע הדורש תגובה מיידי, התוכנית הראשית מופסקת, מופעלת תת-תוכנית (**פעולת פסיקה** או שגרת פסיקה) לזמן קצר, ובסופו של פרק זמן זה השליטה חוזרת לתוכנית הראשית.

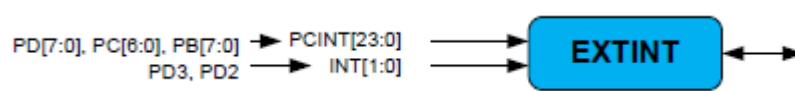
במקרה שלנו, התוכנית הראשית היא הפעולה `loop()`.

5.2 בקר הפסיקות במיקרו-בקר ATmega328P

באיור 3.1 מוצגת יחידת הבקרה של פסיקות החומרה החיצוניות במיקרו-בקר.

לפי האיור יש שתי קבוצות של מבואות הגורמות לפסיקה:

- INT[1:0] – שני הדקים "יעודיים שבאמצעותם אפשר לדווח על אירוע המחייב תגובה מיידי. כל הדק מפעיל שגרת פסיקה משלו.
- PCINT[23:0] 24 הדקים שאפשר לדווח באמצעותם על אירוע של שינוי ברמה הלוגית. קבוצת הדקים זו מחולקת לשלוש תת-קבוצות הדקים. כל תת-קבוצה כזאת מפעילה שגרת פסיקה אחת המשותפת לכל ההדקים.



איור 3.1 בקר הפסיקות במיקרו-בקר

בקר הפסיקות מסוגל לזהות בקשות פסיקה לפי שלוש אפשרויות:

- **עלייה** (rising) – האות החשמלי משנה את ערכו מרמה לוגית 0 לרמה לוגית 1.
- **ירידה** (falling) – האות החשמלי משנה את ערכו מרמה 1 לרמה 0.
- **רמה נמוכה** (low level) – האות החשמלי מוחזק ברמה נמוכה 0.

אם הדק מוגדר פלט (output), שינוי במצבו עלול לעורר בקשת פסיקה. אופציה זו מאפשרת הפעלת שגרת פסיקה גם באמצעות הוראה בתוכנה.

בפרק זה נדון בקבוצת הפסיקה הראשונה INT[1:0] בלבד.

שם הפסיקה	מספר הדק בלוח	הדק במיקרו-בקר
INT0	2	PD2
INT1	3	PD3

טבלה 5.1 מיפוי הדקי הפסיקה

5.3 פעולות פסיקה בסביבת הפיתוח

כאמור, כאשר מתרחש אירוע פסיקה חומרתי בהדק 2 או בהדק 3, מופעלת בלוח הפיתוח שגרת פסיקה (ISR – Interrupt Service Routine) המתאימה להדק.

פעולת attachInterrupt

כותרת הפעולה:

```
void attachInterrupt(pin, ISR, mode)
```

פעולה זו קובעת את אופן קבלת הפסיקה בהדק ואת שגרת הפסיקה שתופעל כאשר יתרחש אירוע פסיקה.

פעולת attachInterrupt מקבלת שלושה פרמטרים:

- pin – מספר הדק הפסיקה
מטעמי תאימות מומלץ להשתמש במקור² digitalPinToInterrupt, המאפשר להמיר את מספר ההדק בלוח הפיתוח למספרו במיקרו-בקר.
- ISR – שם שגרת הפסיקה
פעולה חסרת פרמטרים שאינה מחזירה ערך.
- mode – האופן שבו תתקבל שגרת הפסיקה:

² מקור: פעולה המתבצעת בזמן הידור תוכנית.

אופן	תיאור
low	כאשר ההדק ברמה נמוכה
rising	כאשר יש שינוי שטיבו עלייה
falling	כאשר יש שינוי שטיבו ירידה
change	כאשר יש שינוי

דוגמה: הזימון

```
attachInterrupt(digitalPinToInterrupt(2), myF, RISING);
```

קובע שעם עליית אות המתח בהדק 2 בלוח הפיתוח תופעל שגרת פסיקה myF.

פעולת detachInterrupt

כותרת הפעולה:

```
void detachInterrupt(pin)
```

פעולה זו חוסמת את קבלת הפסיקה הנתונה.

פעולת detachInterrupt מקבלת פרמטר יחיד:

- pin – מספר הדק הפסיקה

מטעמי תאימות מומלץ להשתמש במקרו digitalPinToInterrupt, המאפשר להמיר

את מספר ההדק בלוח הפיתוח למספרו במיקרו-בקר.

פעולת noInterrupts

כותרת הפעולה:

```
void noInterrupts()
```

פעולה זו מונעת קבלת כל פסיקה.

שימו לב! שימוש בפסיקה יפסיק את פעילותן התקינה של פעולות התלויות במנגנון הפסיקה,

כמו למשל delay.

פעולת interrupts

כותרת הפעולה:

```
void interrupts()
```

פעולה זו מאפשרת את כל הפסיקות.

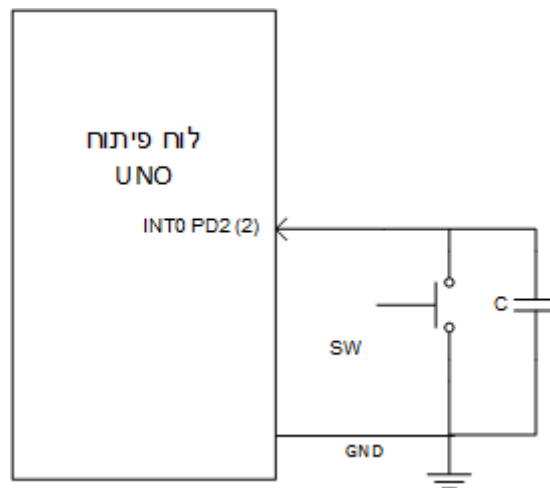
פעולת volatile

מילת מפתח המופיעה בהגדרת משתנה. מילת מפתח מאלצת את המהדר להגדיר את המשתנה במרחב זיכרון ה-RAM ולא בזיכרון זמני (אוגרים). חשוב להגדיר את המשתנים הגלובליים בתוכנית באמצעות מילת מפתח זו אם נעשה בהם שימוש בשגרת פסיקה.

דוגמה 5.2: הדלקת נורת לד בקצב משתנה לפי לחצן

כתבו תוכנית הגורמת להבהוב של דיודת הלד המחוברת להדק 13. זמן ההפעלה והכיבוי של נורת הלד יהיה תלוי בלחצן המחובר בתצורת Pull-Up להדק 2. כל לחיצה תחליף את קצב ההפעלה והכיבוי. הקצב הדרוש הוא: 500mSec ו-1000mSec.

באיור 3.2 נראה המעגל החשמלי לחיבור הלחצן אל לוח הפיתוח. מחברים מפסק בין הדק 2 ובין האדמה, מכיוון שבחרנו להשתמש בנגד Pull-Up המובנה במיקרו-בקר. הקבל שחובר במקביל למפסק הלחיצה נועד למנוע ריטוטים.



איור 5.2 חיבור מפסק להדק פסיקה

```
const int ledPin= 13;           // הגדרת קבוע עבור הדק הled
const int interruptPin= 2;      // הגדרת קבוע עבור הדק המפסק במבוא הפסיקה
volatile byte state = LOW;     // משתנה המציין את מצב הלחיצה
volatile int timedelay;        // הגדרת משתנה לקביעת זמן ההפעלה והכיבוי

void setup() {
    pinMode(ledPin, OUTPUT);    // הגדרת הדק הled כמוצא
    pinMode(interruptPin, INPUT_PULLUP); // Pull Up כמבוא
    attachInterrupt(digitalPinToInterrupt(interruptPin),myISR, RISING);
    timedelay = 500;           // קביעת זמן ההשהיה ההתחלתי
}

void loop() {
    digitalWrite(ledPin, HIGH); // הדלקת דיודת הled
    delay(timedelay);
    digitalWrite(ledPin, LOW);   // כיבוי דיודת הled
    delay(timedelay);
}

void myISR(){                   // שגרת הפסיקה
    state = !state;             // החלפת מצב הלחיצה
    if (state == LOW)           // עדכון זמן ההשהיה לפי מצב הלחיצה
        timedelay = 500;
    else
        timedelay = 1000;
}
```

הסבר התוכנית

בקטע הראשון של התוכנית הוגדרו קבועים ומשתנים כמפורט להלן

שם הקבוע או המשתנה	תיאור
ledPin	קבוע מספרי שלם המציין את מספר ההדק בלוח עבור הled
interruptPin	קבוע מספרי שלם המציין את מספר ההדק בלוח עבור מבוא פסיקת לחצן
state	משתנה מספרי שלם בגודל בית מהגדרה volatile עבור מצב הלחיצה
timedelay	משתנה מספרי שלם מהגדרה volatile עבור זמן ההשהיה

פעולת setup

- הגדרת הדק הלד כפלט
- הגדרת הדק הפסיקה כמבוא מסוג Pull-Up
- פעולת attachInterrupt המגדירה את הדק הפסיקה להדק 2, את אופן קבלת הפסיקה לעלייה ואת שגרת הפסיקה myISR שתופעל בעת אירוע הפסיקה
- הגדרת זמן ההשהיה ההתחלתי ל-500mSec.

פעולת loop

פעולת הדלקה וכיבוי נורת הלד לסירוגין לפי זמני ההשהיה המוגדרים במשתנה timedelay.

שגרת פסיקה myISR

פעולה המשנה את ערך המשתנה state ואז מעדכנת את המשתנה timedelay לאחד מהערכים שנקבעו.

שגרת הפסיקה מופעלת בכל עליית אות המתח בהדק הפסיקה, כלומר השינוי יחול בכל פעם שמשחררים את הלחצן.



תרגול

שאלה 5.1

מעוניינים לשפר את התוכנית מדוגמה 5.2 כך שבמקום שההשהיה תהיה 500mSec או 1000mSec היא תיקבע על פי ערכו של מתח המתקבל מפוטנציומטר.

כתבו תוכנית שתציג בפעולה loop את ערך המתח המתקבל במבוא תקבילי A1. לאחר בחירת מתח רצוי המשתמש ילחץ על הלחצן. בכל לחיצה יעודכן זמן ההשהיה על פי ערכו הבינארי הגולמי (לפני ההמרה) של אות המתח, והנורה תהבהב בקצב הדרוש.

מומלץ לקבוע את קצב התקשורת הטורית ל-115200bps.

5.4 פסיקות מונה

בפרק הקודם תיארנו את המבנה הפנימי של המיקרו-בקר. למיקרו-בקר שלושה מונים המשמשים גם כקוצבי זמן. שלושת המונים משמשים את סביבת הפיתוח IDE לצורך פעולות כמו השהייה ויצירת אותות PWM המשמשים כיציאה אנלוגית.

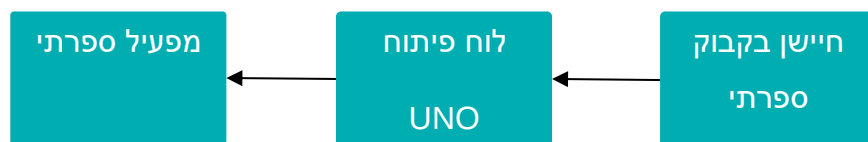
מכיוון שאין לנו אפשרות³ לגשת למונים של המיקרו-בקר, ניצור מונים באמצעות מבוא פסיקה. בכל פעם שתתרחש פסיקה נקדם משתנה גלובלי באחד או נחסיר ממשתנה גלובלי אחד.

מונה משמש לספירת אירועים ממקור חיצוני.

דוגמה 5.3: ספירת בקבוקים העוברים בפס ייצור

במפעל למילוי בקבוקי מים מעבירים את הבקבוקים לאחר המילוי במסוע חשמלי אל עמדת האריזה. לפני הגעת הבקבוקים לעמדת האריזה סופרים את הבקבוקים, וכל 12 בקבוקים מופרדים באמצעות מנגנון אלקטרו-מכני לצורך אריזתם יחד.

באיור 5.3 מוצג תרשים מלבנים עקרוני למערכת הכוללת חיישן ספרתי המזהה מעבר של בקבוק במסוע ומפעיל ספרתי הפוקד על המנגנון האלקטרו-מכני. בקרת התהליך מיושמת באמצעות ערכת פיתוח מסוג UNO.



איור 5.3 מערכת בקרת התהליך

חיישן הבקבוק מחובר להדק 3 בלוח הפיתוח והמפעיל הספרתי מחובר להדק 8 בלוח הפיתוח. על תוכנית ההפעלה למנות את הבקבוקים. כאשר יגיע מספר הבקבוקים ל-12 יישלח דופק רגעי ברוחב 20mSec למנגנון האלקטרו-מכני.

³ מובן שאפשר לגשת למונים באמצעות מילות בקרה וערך מתאימים, אולם אפשרות זו חורגת מתחום הידע הנדרש בתוכנית הלימודים.

תוכנית

```
const int operatePin= 8;           // הגדרת קבוע עבור הדק המפעיל האלקטרו-מכני
const int interruptPin= 3;        // הגדרת קבוע עבור הדק החיישן במבוא הפסיקה
volatile int counter;             // הגדרת משתנה המשמש כמונה
void setup() {
    digitalWrite(operatePin, LOW); // קביעת מצב התחלתי למפעיל
    pinMode(operatePin, OUTPUT);   // הגדרת ההדק כמוצא
    pinMode(interruptPin, INPUT);  // הגדרת הדק החיישן כמבוא
    attachInterrupt(digitalPinToInterrupt(interruptPin),myCnt, RISING);
    counter = 0;                   // איפוס המונה
}
void loop() {
}
void myCnt(){                      // שגרת הפסיקה
    counter++;                     // קידום המונה
    if (counter == 12) {           // בדיקה אם מספר הבקבוקים הוא 12
        counter = 0;              // איפוס המונה
        digitalWrite(operatePin, HIGH); // יצירת דופק קצר
        delay(10);
        digitalWrite(operatePin, LOW);
    }
}
```

הסבר התוכנית

בקטע הראשון של התוכנית הוגדרו קבועים ומשתנים כמפורט להלן

שם הקבוע או המשתנה	תיאור
operatePin	קבוע מספרי שלם המציין את מספר ההדק בלוח עבור המפעיל האלקטרו-מכני
interruptPin	קבוע מספרי שלם המציין את מספר ההדק בלוח עבור מבוא פסיקת חיישן
Counter	משתנה מספרי שלם מהגדרה volatile עבור המונה

פעולת setup

- הגדרת הדק המפעיל כפלט והגדרת ערכו ההתחלתי
- הגדרת הדק הפסיקה עבור החיישן
- פעולת attachInterrupt המגדירה את הדק 3 כהדק הפסיקה, את אופן קבלת הפסיקה לעלייה ואת שגרת הפסיקה myCnt שתופעל בעת אירוע הפסיקה
- הגדרת ערכו ההתחלתי של המונה ל-0.

פעולת loop

הפעולה ריקה. כל התהליך מתבצע בשגרת הפסיקה.

שגרת פסיקה myCnt

הפעולה משנה את ערכו של המשתנה counter ואז בודקת את ערכו. אם הערך שווה ל-12, המשתנה מאופס ונשלח למפעיל דופק ברוחב זמן 10mSec.

שגרת הפסיקה מופעלת בכל עליית אות המתח בהדק הפסיקה, כלומר בכל פעם שבקבוק עובר.



שאלה 5.2

למבנה שני פתחים, אחד מהם משמש לכניסה בלבד והשני משמש ליציאה בלבד. בכל כניסה מוצב חיישן המזהה מעבר של אדם. הפתחים מסודרים כך שרק אדם אחד יכול לעבור בפתח, ולכן הזיהוי יהיה של אדם אחד.

נניח כי בתחילה המבנה ריק.

מתכננים באמצעות לוח הפיתוח UNO מערכת לספירת האנשים הנכנסים והיוצאים מהמבנה. אם אין אנשים במבנה (מספר האנשים הוא 0), תינתן הוראה לכבות את התאורה במבנה. אם יש לפחות אדם אחד – התאורה תפעל.

- א. זהו את סוגי הכניסות והיציאות למערכת המתוארת (ספרתית או תקבילית).
- ב. תארו את המערכת בתרשים מלבנים.
- ג. הסבירו את שתי גישות הפתרון האפשריות למערכת המתוארת.
- ד. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת התשואל. בפתרון התייחסו להקצאת ההדקים.
- ה. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת הפסיקה. בפתרון התייחסו להקצאת ההדקים.

מוסיפים פתח יציאה שני למבנה.

1. אילו שינויים נדרשים בפתרון לפי גישת התשאול?
2. אילו שינויים נדרשים בפתרון לפי גישת הפסיקה?

שאלה 5.3

מתכננים שעון שניות חד-ספרתי הסופר מטה מערך 5. בתחילת כל הפעלה השעון מאותחל לערך ההתחלתי (5). באמצעות מפסק לחיצה START תחל ספירה מטה. כאשר השעון מגיע לערך 0 יופעל זמזם למשך שנייה אחת והשעון יאותחל לערכו ההתחלתי.

- כדי לשנות את ערך השעון מספקים למערכת אות גל ריבועי בתדר של 1Hz ממקור חיצוני (מחולל אותות).
- השעון יוצג באמצעות יחידת תצוגה של שבעה מקטעים (7-SEGMENT) מסוג CA.

- א. זהו את סוגי הכניסות והיציאות למערכת המתוארת (ספרתית או תקבילית).
- ב. תארו את המערכת בתרשים מלבנים.
- ג. הסבירו את שתי גישות הפתרון האפשריות למערכת המתוארת.
- ד. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת התשאול. בפתרון התייחסו להקצאת ההדקים.
- ה. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת הפסיקה. בפתרון התייחסו להקצאת ההדקים.

שאלה 5.4

נתונה התוכנית שלהלן:

```
const int interruptPin= 3;
volatile int counter;
void setup() {
    pinMode(interruptPin, INPUT);
    Serial.begin(9600);
}
void loop() {
    counter = 0;
    attachInterrupt(digitalPinToInterrupt(interruptPin),myIsr, RISING);
    delay(1000);
    detachInterrupt(digitalPinToInterrupt(interruptPin));
    Serial.println(counter);
}
void myIsr(){
    counter++;
}
```

- א. הסבירו את השורה המגדירה את המשתנה counter.
- ב. הסבירו מה מתבצע בפעולה setup.
- ג. מהו התנאי לכך ששגרת הפסיקה תתבצע?
- ד. מה מתבצע בשגרת הפסיקה?
- ה. הסבירו כל שורה בתוכנית הראשית.
- ו. מחברים להדק הפסיקה אות גל ריבועי בתדר 100Hz. מה תציג התוכנית?
- ז. מה מבצעת התוכנית?

שאלה 5.5

נתונה התוכנית שלהלן:

```
const int interruptPin= 2;
const int out1 = 7;
const int out2 = 6;
volatile int counter;
void setup() {
    pinMode(interruptPin, INPUT);
    counter = 0;
    attachInterrupt(digitalPinToInterrupt(interruptPin),myIsr, RISING);
    pinMode(out1, OUTPUT);
    pinMode(out2, OUTPUT);
    tone(out2, 1000);
}
void loop() {
}
void myIsr(){
    counter++;
    if (counter == 200) {
        counter = 0;
        digitalWrite(out1, !digitalRead(out1));
    }
}
```

נתון שהדק 2 והדק 6 מחוברים זה לזה.

- א. הסבירו את השורה המגדירה את המשתנה counter.
- ב. הסבירו מה מתבצע בפעולה setup.
- ג. מה מבצעת הפעולה tone? למה היא משמשת?
- ד. מהו התנאי לכך ששגרת הפסיקה תתבצע?
- ה. מה מתבצע בשגרת הפסיקה?
- ו. מה מבצעת התוכנית?

סיכום פרק 5

בפרק 5 למדתם את הנושאים האלה:

1. הגדרת תוכנית סדרתית
2. הפעלת תת-תוכניות לפי עיקרון של "בו זמניות"
3. תוכניות לפי גישת התשאול (Polling)
4. תוכניות לפי גישת הפסיקה (Interrupt)
5. בקר הפסיקות במיקרו-בקר
6. מבואות פסיקה INT0, INT1
7. אופני קבלת פסיקה: עלייה, ירידה ושינוי
8. שגרת פסיקה
9. פעולת attachInterrupt הקושרת את שגרת הפסיקה להדק וקובעת את אופן קבלת הפסיקה
10. פעולת detachInterrupt להפסקת קבלת פסיקה
11. פעולת חסימת כל הפסיקות noInterrupts
12. פעולת אפשרור כל הפסיקות interrupts
13. פעולות עם מונים
14. שימוש בפסיקות חומרה למימוש רעיון המונים

פרק 6: תקשורת טורית UART בין מיקרו-בקר למחשב

תקשורת טורית היא תהליך של העברת סיבית אחת בכל פעם, באופן סדרתי, דרך ערוץ תקשורת יחיד. בתקשורת מקבילית, לעומת זה, מועבר נתון באמצעות כמה סיביות כיחידה אחת בו בזמן דרך כמה ערוצי תקשורת.

השימוש בתקשורת טורית נפוץ בקווי תקשורת ארוכים וברשתות תקשורת מחשבים. עקב ההתפתחות הטכנולוגית בתחום זה, כיום תקשורת טורית נפוצה גם במרחקים קצרים.

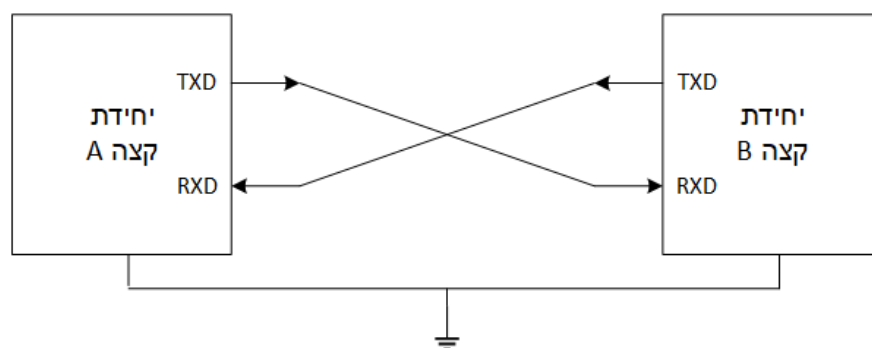
6.1 תקן TIA232

אחד מתקני התקשורת הטורית הוותיקים הוא תקן RS232, שהוצג לראשונה בשנת 1962. כיום יש גרסת עדכון F לתקן זה, והוא נקרא TIA-232. במקור נועד תקן זה לחבר בין ציוד תקשורת (מודם) לציוד קצה (מחשב, מסוף). בפועל השתמשו בו גם לחיבור שתי יחידות קצה זו לזו (חיבור בין שתי מערכות ממוחשבות). ברוב המחשבים האישיים אין חיבורי RS232, וכדי להשתמש בתקן זה יש להשתמש במתאם USB-to-RS232. בלוחות Arduino יש לפחות ערוץ אחד של תקשורת טורית, עם הדקים ייעודיים לכך, ויש אפשרות לדמות תקשורת טורית בתוכנה (SoftawreSerial) בהדקים אחרים.

קווי החיבור

תקן EIA232 מגדיר את קווי החיבור בין יחידות הקצה. קווי החיבור על פי התקן כוללים התייחסות לממשק בין ציוד התקשורת לציוד הקצה. ממשק החיבור לציוד התקשורת חורג מהידע הנדרש בפרק זה, ולכן לא נתייחס לכל קווי החיבור. בפרק זה נתמקד בקווי החיבור ההכרחיים לקשר בין שתי מערכות ממוחשבות.

באיור 6.1 נראה חיבור בין שתי יחידות ממוחשבות.



תפקידי ההדקים

קו שיחידת קצה משדרת דרכו	Transmit Data	TXD
קו שיחידת קצה קולטת דרכו	Receive Data	RXD

המידע המשודר על ידי יחידת קצה אחת בקו TXD נקלט על ידי יחידת הקצה האחרת בקו RXD. חיבור כזה נקרא חיבור מוצלב. כיוון שיש ערוץ נפרד לשידור TXD וערוץ נפרד לקליטה RXD, אפשר לקיים שידור דו-כיווני במקביל. שיטה זו של שידור נקראת Full Duplex.

רמות המתח

תקן EIA232 מגדיר את רמות המתח בקווי התקשורת עבור הרמות הלוגיות. התקן מתייחס לחיבור קווי תקשורת ארוכים. רמות המתח האלה אינן רלוונטיות לפרק זה מכיוון שאנו נחבר את יחידת הקצה בחיבור ישיר ובקווים קצרים שבהם הרמות הלוגיות התקניות יפעלו בצורה תקינה. הרמות התקניות נקראות רמות TTL:

ערך לוגי 0V–0 וערך לוגי 1V–5.

סוגי תקשורת

תקן EIA232 תומך בשתי שיטות שידור, סינכרונית וא-סינכרונית. בשיטת השידור הסינכרונית יש גם קווי אות שעון נפרדים עבור המשדר ועבור המקלט. אותות השעון נועדו לתאם בין הצד המשדר לצד הקולט. בתקשורת א-סינכרונית לא מסופק אות שעון ולכן נדרשת שיטה שונה לתיאום בין המשדר למקלט. בשיטה הא-סינכרונית קו השידור הוא במצב לא פעיל, ואז הוא עובר זמנית למצב המציין תחילת שידור. המידע מועבר לאחר ציון תחילת השידור. בסוף השידור חוזר קו השידור למצב לא פעיל. תקשורת א-סינכרונית מתאפיינת בשידור במבנה של מסגרות – Frames. רוב מערכות המחשבים, ובכללן המחשב האישי, תומכות בתקשורת א-סינכרונית בלבד, ובפרק זה נדון רק בה.

מסגרת שידור

בתקשורת א-סינכרונית המידע מועבר בין יחידות הקצה לפי הצורך, כלומר רק כאשר יש מידע להעברה מתחיל תהליך של שידור. כאשר אין מידע לשידור, קו השידור הוא במצב לא פעיל. מצב קו לא פעיל מתאפיין ברמה לוגית 1.

המידע להעברה מפוצל, לפי הצורך, ליחידות מידע. כל יחידת מידע משודרת לחוד, במבנה שנקרא מסגרת מידע. תקן RS232 תומך בהעברת יחידות מידע בגודל 5–9 סיביות. ברוב היישומים נהוג היום להשתמש ב-8 סיביות מידע. תקן RS232 אינו מתייחס לתקינות העברת

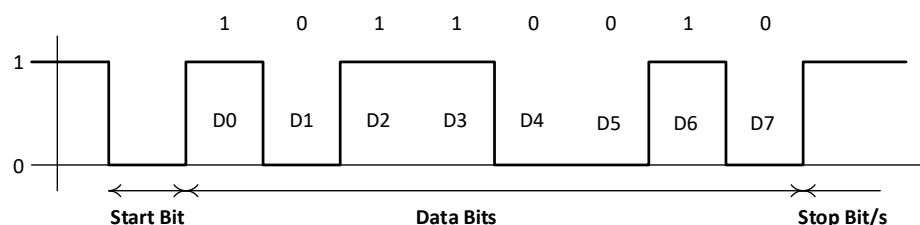
המידע כולו, אולם נותן מענה בסיסי בלבד לזיהוי התקינות של העברת יחידת מידע. תקינות העברת המידע נבדקת בשיטת בדיקת זוגיות – Parity Check.

כל מסגרת שידור מתחילה במעבר ממצב לא פעיל – 1 – למצב פעיל – 0. מצב זה נשמר זמן קצוב המאפיין את זמן השידור של סיבית בודדת. סיבית זו נקראת סיבית התחלה (Start bit). לאחר סיבית ההתחלה משודרות זו אחר זו הסיביות של יחידת המידע. תחילה משודרת הסיבית המשמעותית פחות (LSB), ולבסוף משודרת הסיבית המשמעותית יותר (MSB). לאחר סיביות המידע אפשר להוסיף סיבית ביקורת. שידור המסגרת מסתיים לאחר החזרת הקו למצב לא פעיל למשך זמן של סיבית אחת לפחות. הסיבית האחרונה נקראת סיבית הסיום – אות סיבית העצירה (Stop bit). אפשר שיהיו כמה סיביות עצירה.

דוגמה 6.1: שידור התו 'M'

באיור 10.7 מוצגת העברת התו 'M' בשימוש ב-8 סיביות מידע וללא סיבית ביקורת. לתו 'M' קוד ASCII, שערך $77=4DH=01001101B$.

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	0	1	1	0	1



איור 6.2 שידור התו 'M' ביחס לזמן

קצב תקשורת Baud Rate

אחד ממאפייני התקשורת החשובים הוא מהירות העברת המידע בין יחידות הקצה. מאפיין זה קובע את כמות המידע שאפשר להעביר בשנייה אחת. כיוון שהמידע מועבר במבנה של מסגרות שאורכן אינו אחיד, מודדים את הקצב במספר הסיביות המשודרות בשנייה אחת. יחידות הקצב נקראות סל"ש (סיביות לשנייה), או bps (bit per second). בתקן RS232 מקובלות היחידות baud לתיאור קצב הסיביות.

ערכי קצב התקשורת המקובלים:

110, 150, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600.

זמן סיבית מחושב כך:

$$t_{BIT} = \frac{1}{baud} [Sec]$$

זמן מסגרת מחושב כך:

$$t_{FRAME} = n \times t_{BIT} = n \times \frac{1}{baud} = \frac{n}{baud} [Sec]$$

n הוא אורך המסגרת בסיביות.

דוגמה 6.2: חישובי זמן סיבית וזמן מסגרת

ערוץ תקשורת משדר בתקן RS232 בקצב baud=4800, 8 סיביות מידע וסיבית עצירה אחת. משדרים את המידע ללא סיבית ביקורת.

- א. חשבו את זמן הסיבית.
- ב. חשבו את מספר הסיביות במסגרת מידע.
- ג. חשבו את זמן המסגרת.

פתרון

$$t_{BIT} = \frac{1}{baud} = \frac{1}{4800} = 208.3 \mu Sec$$

א. מספר הסיביות במסגרת מידע:

$$10 = 1 \text{ סיבית עצירה} + 8 \text{ סיביות מידע} + \text{סיבית התחלה} = n$$

$$t_{FRAME} = \frac{n}{baud} = \frac{10}{4800} = 2.083 mSec$$

זמן המסגרת קובע כמה תווים (תו – אות) אפשר לשדר ברציפות במשך שנייה אחת:

$$CharRate = \frac{1}{t_{FRAME}} [cps]$$

שאלה 6.1

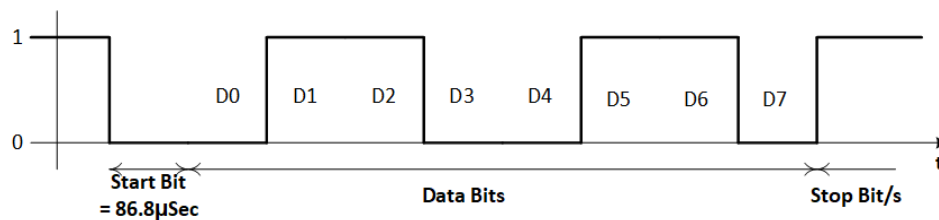
משדרים את התו 'א' לפי הפרמטרים האלה:

8 סיביות מידע, סיבית עצירה אחת, ללא סיבית ביקורת ובקצב 19200bps.

- א. מהו הקוד של התו 'א' לפי בסיס עשרוני, הקסדצימלי ובינארי? היעזרו במנוע חיפוש.
- ב. שרטטו דיאגרמת שידור עבור התו ביחס לזמן. ציינו בדיאגרמה את ערכי הרמות הלוגיות ואת ערכי הזמנים.
- ג. מהו זמן המסגרת?
- ד. כמה תווים אפשר לשדר ברציפות על פי הפרמטרים של התקשורת?

שאלה 6.2

נתונה דיאגרמת השידור שלהלן:



- א. מהו הערך הבינארי המשודר? הציגו גם בבסיס 10 וגם בבסיס 16.
- ב. מהו התו המשודר? היעזרו במנוע חיפוש.
- ג. חשבו את קצב השידור.
- ד. חשבו את זמן המסגרת.
- ה. כמה תווים אפשר לשדר ברציפות על פי הפרמטרים של התקשורת?

6.2 תקשורת טורית UNO

קווי התקשורת בלוח

באיור 6.3 מסומנים הדקי התקשורת TX ו-RX בלוח.



איור 6.3 קווי תקשורת בלוח UNO

קווי התקשורת האלה משמשים לכמה צרכים:

- תהליך העלאת (Upload) התוכנית ללוח
- העבודה מול ה-Serial Monitor בזמן ניפוי שגיאות (Debug)
- חיבור התקנים ללוח עבור יישום חומרה של המשתמש.

השימוש בתקשורת טורית RS232 מוגבל לחיבור פעיל אחד בלבד, כלומר אין אפשרות להשתמש באותם קווים לחיבור התקנים שונים באותו זמן. לאחר תהליך ההעלאה קווי התקשורת פנויים לשימוש בתוכנת Monitor או להתקני חומרה. אם בשלב ההעלאה יש התקן חומרה המחובר לקווי התקשורת, עלולה להיות התנגשות וההעלאה תיכשל. במקרה כזה יש לנתק את קווי התקשורת של רכיב החומרה.

אפשרות אחרת היא לחבר את התקן החומרה להדקים אחרים בלוח ולהשתמש במחלקה אחרת, SoftwareSerial, המדמה תקשורת טורית. נדגים את השימוש במחלקה זו בהמשך.

המחלקה Serial

בפרק הראשון הכרנו את המחלקה Serial ושלוש פעולות שלה. בחלק זה נרחיב את ההיכרות עם מחלקה זו. בטבלה 6.1 שלהלן מתוארות חלק מפעולות מחלקה זו (UML).

הפעולה	תיאור הפעולה
<pre>void begin(long speed) void begin(long speed, int config)</pre>	אתחול התקשורת למהירות המוגדרת בפרמטר speed והפרמטרים של התקשורת המוגדרים בפרמטר config. ברירת המחדל היא SERIAL_8N1, 8 סיביות מידע, ללא סיבית ביקורת ועם סיבית סיום אחת.
<pre>void end()</pre>	הפסקת השימוש בהדקים RX-ITX למטרות תקשורת. לאחר פעולה זו אפשר להשתמש בהדקים למטרות קלט/פלט ספרתי.
<pre>size_t print(val) size_t print(val, format)</pre>	שליחת val כמידע טקסטואלי דרך ערוץ התקשורת: • ערך שלם יוצג כמספר שלם. • ערך ממשי יוצג בדיוק של שתי ספרות אחרי הנקודה. • תו או מספר כבית יוצג כתו. • מחרוזת תוצג כמחרוזת. הפרמטר format קובע את אופן ההצגה של ערכים מספריים לפי בסיסים שונים או קביעת מספר הספרות אחרי הנקודה. הפעולה print מציגה את המידע בהמשך השורה וההוראה println מציגה את המידע ועוברת לשורה חדשה. הפעולה מחזירה את מספר הבתים שנכתבו. size_t מוגדר מספר שלם חיובי. הערה: כדי לשלוח נתון מספרי באופן בינארי השתמשו בפעולה write.
<pre>size_t write(byte val) size_t write(string str) size_t write(byte buf[], int len)</pre>	שידור המידע באופן בינארי. val נתון מסוג בית. str נתון שהוא מחרוזת תווים. buf מערך של בתים באורך בפרמטר len. הפעולה מחזירה את מספר הבתים שנכתבו. size_t מוגדר מספר שלם חיובי.
<pre>int read()</pre>	קריאה והחזרה של הערך שנקלט בערוץ התקשורת הטורית. הפעולה תחזיר 1- אם לא קיים מידע זמין.
<pre>int available()</pre>	החזרת מספר הבתים שנקלטו בערוץ התקשורת הטורית.

לערוץ התקשורת הטורית חוצץ (buffer) בגודל 64 בתים.	
חיפוש מספר שלם תקין והחזרתו. אפשר לציין תו המשמש להתעלמות מסימנים במספר שלם. למשל סימן >, < המציין הפרדה בין קבוצת ספרות המציינות אלפים: 1,000,000. • דילוג על סימנים בהתחלה שאינם סימן מינוס או ספרה. • אם נקלטה ספרה ראשונה, ההמרה מסתיימת אם חלף זמן קצוב או שמופיע תו שאינו מספר. • אם לא נקלטה ספרה בזמן קצוב – יוחזר ערך 0.	long parseInt() long parseInt(char skipChar)
קריאה מחוצץ הקלט למערך תווים char[] או מערך בתים byte[]. הפעולה מסתיימת אם נקלטו len בתים או אם עבר זמן קצוב. אם משתמשים בגרסה Until הקלט יסתיים כאשר זוהה התו c או שנקלטו len בתים או עבר זמן קצוב. אם הקלט לא תקין – יוחזר הערך 0.	size_t readBytes(buffer, int len) size_t readBytesUntil(char c, buffer, int len)

טבלה 6.1 תרשימים חלקי של פעולות המחלקה

קלט מפתח טורי

כדי לקלוט מידע מערוץ טורי נשתמש בפעולה read(), המחזירה את ערכו של המידע שנקלט אם קיים מידע כזה. אם לא קיים ערך, יוחזר הערך המספרי 1-. השימוש בפעולה ללא בדיקה מוקדמת אם קיים ערך זמין יאלץ אותנו לבדוק אם הערך הנקלט שונה מ-1:-

```
int val;           // משתנה עבור הערך הנקלט
// קטע תוכנית לקלט
val = Serial.read(); // קלט נתון
if (val != -1) {    // בדיקה אם קיים נתון
    // TODO          הוראות אם יש נתון חדש
}
```

כדי לקרוא מידע רק אם יש קלט זמין נשתמש בפעולה available(), המחזירה את מספר הבתים של מידע זמין בחוצץ הקלט שגודלו 64 בתים. אם מספר הבתים הזמינים בחוצץ גדול מ-0, אפשר לבצע קריאה של המידע ולהשתמש בו.

```
int val;           // משתנה עבור הערך הנקלט
// קטע תוכנית לקלט
while (Serial.available() > 0) { // בדיקה אם קיים נתון בחוצץ הקלט
    val = Serial.read(); // קלט נתון
    // TODO           הוראות אם יש נתון חדש
}
```

השימוש בלולאת while מאפשר טיפול רציף במידע הנקלט. התנאי `(Serial.available() > 0)` בודק אם מספר הבתים גדול מ-0, כלומר אם יש לפחות נתון אחד הממתין לקריאה.

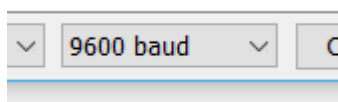
אפשר לכתוב את התנאי בצורה מקוצרת `(Serial.available())`, מכיוון שבשפה זו, בניגוד לשפת C#, לא חייבים לכתוב ערך בוליאני `true` או `false`. בשפה זו הערך המספרי השלם 0 מייצג ערך שקר – `false` וכל ערך מספרי שונה מממנו מייצג ערך אמת – `true`.

דוגמה 6.3: קלט בית read() ופלט ב-Serial Monitor

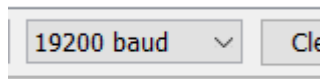
כתבו תוכנית הקולטת טקסט מה-Monitor ומציגה את המידע הנקלט.

```
int val ;           // משתנה לקריאת הערך מהמפתח הטורי
void setup() {
    Serial.begin(19200); // אתחול תקשורת לקצב 19200
    Serial.println("Enter a number");
}
void loop() {
    while (Serial.available()>0){ // בדיקה אם יש נתון ממתין
        val = Serial.read(); // קריאת ערך הנתון
        Serial.print("You entered ");
        Serial.print(val, BIN); // הצגת הערך בבסיס בינארי
        Serial.print("B, ");
        Serial.print(val, HEX); // הצגת הערך בבסיס הקסדצימלי
        Serial.print("H, ");
        Serial.println(val); // הצגת הערך בבסיס עשרוני
        Serial.println("Enter a number");
    }
}
```

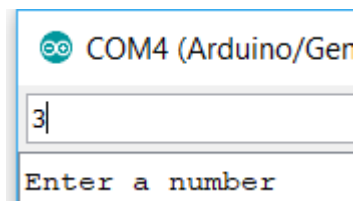
שימו לב! קצב התקשורת נקבע על 19200. ברירת המחדל בתוכנת המוניטור היא 9600.



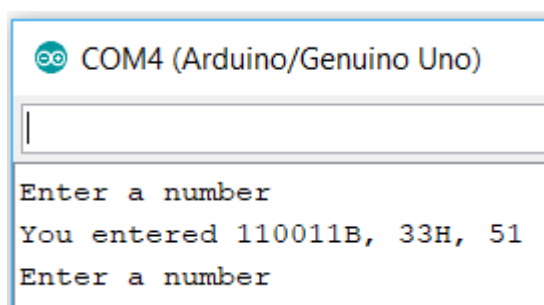
יש לקבוע את הקצב גם בתוכנת המוניטור:



לאחר קביעת הקצב הנכון תציג תוכנת המוניטור את ההודעה Enter a number. נזין ערך בשורת שליחת המידע:



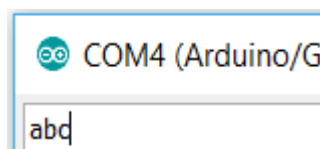
לחיצה על מקש <Enter? או לחיצה על כפתור Send תגרור את התגובה:



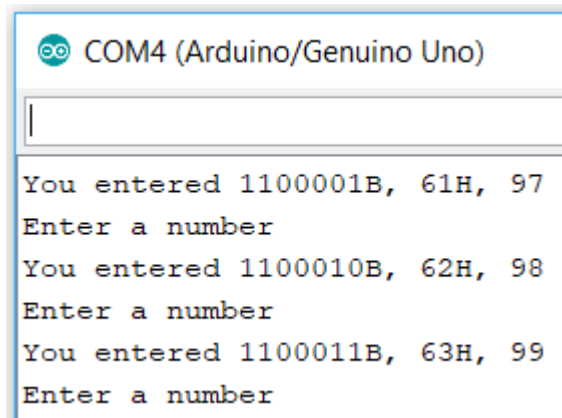
התו 3 אינו מופיע, אלא ערכו לפי קוד ASCII בבסיסים השונים.

המשמעות של עובדה זו היא שתוכנת המוניטור שולחת תו באופן בינארי ואילו הפעולה read() קולטת אותו כמספר רגיל ולא כתו 3.

מה יקרה אם נשדר את המחרוזת abc?



התווים ישודרו באופן עוקב זה אחר זה ויוצגו כמספרים זה אחר זה!

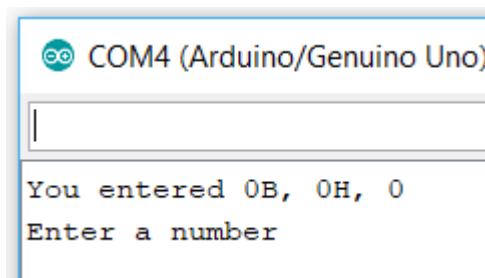


דוגמה 6.4: קלט מספר שלם parseInt() ופלט ב-Serial Monitor

כתבו תוכנית הקולטת מספר מה-Monitor ומציגה את המידע הנקלט.

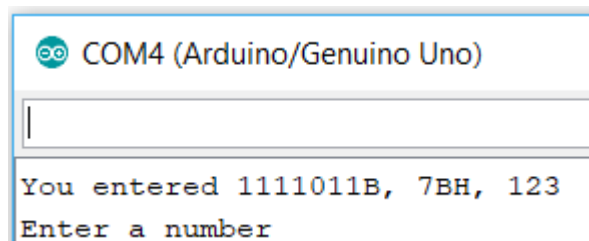
```
int val ; // משתנה לקריאת הערך מהמפתח הטורי
void setup() {
    Serial.begin(19200); // אתחול תקשורת לקצב 19200
    Serial.println("Enter a number");
}
void loop() {
    while (Serial.available() > 0) { // בדיקה אם יש נתון ממתין
        val = Serial.parseInt(); // קריאת ערך מספרי שלם
        Serial.print("You entered ");
        Serial.print(val, BIN); // הצגת הערך בבסיס בינארי
        Serial.print("B, ");
        Serial.print(val, HEX); // הצגת הערך בבסיס הקסדצימלי
        Serial.print("H, ");
        Serial.println(val); // הצגת הערך בבסיס עשרוני
        Serial.println("Enter a number");
    }
}
```

1. דוגמה לשליחת מחרוזת abc במקום מספר



כאשר לא מזוהה ערך מספרי, הפעולה מחזירה 0.

2. דוגמה לשליחת מספר משולב במחרוזת abc123



המספר שמוחזר הוא הערך 123.

דוגמה 6.5: קלט מחרוזת או מערך בתים ופלט ב-Serial Monitor

כתבו תוכנית הקולטת טקסט מה-Monitor ומציגה את המידע הנקלט.

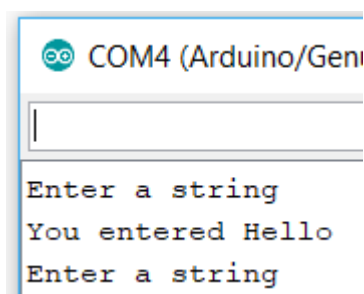
```
#define ARRAYSIZE 20
char val [20];           // מערך תווים
int s;                   // משתנה למספר התווים שנקלטו
void setup() {
    Serial.begin(19200);   // אתחול תקשורת לקצב 19200
    Serial.println("Enter a string");
}
void loop() {
    while (Serial.available() > 0) { // בדיקה אם יש נתון ממתיך
        s = Serial.readBytes(val, ARRAYSIZE); // קריאת מחרוזת תווים
        val[s] = 0; // מוסיפים לסוף המחרוזת ערך 0
        Serial.println(val); // הצגת התווים
        Serial.println("Enter a string");
    }
}
```


בתוכנית הוגדר מערך val בגודל 20 תווים. המשתנה s מציין כמה תווים נקלטו בפועל.

שימו לב לדרך שבה הוגדר קבוע מספרי בשפה זו: #define ARRAYSIZE 20

הפעולה readBytes מקבלת כפרמטרים את מערך התווים val ואת האורך המרבי לקליטה ARRAYSIZE. הפעולה ממתינה לקליטת 20 תווים או עד שיחלוף זמן קצוב מקליטת התו האחרון. אם מתקיים אחד מהתנאים, הפעולה מסתיימת, המשתנה s מתעדכן למספר התווים שנקלטו בפועל והמערך val מכיל את הערכים שנקלטו.

בדוגמה שלהלן הוקלדה המחזורת Hello:



 תרגול

שאלה 6.3

כתבו תוכנית הקולטת סדרת תווים מתוכנת המוניטור ומחזירה את מספר התווים שנקלטו שערכם מתאים לקוד ASCII של התו 'W'.

שאלה 6.4

- א. כתבו תוכנית הקולטת תו מתוכנת המוניטור. התו שנקלט הוא '1'. נורת הLED המחוברת להדק 13 תדלוק, ואם התו שנלקט הוא '0', הנורה תכבה. כל תו אחר לא יגרום שינוי במצב הנורה.
- ב. בדקו את התוכנית בלוח ה-UNO.

שאלה 6.5

- א. כתבו תוכנית הקולטת ערך מספרי שלם מתוכנת המוניטור. אם המספר נמצא בתחום שמ-0 עד 255, יופק אות PWM לפי הערך שהתקבל.
- ב. בדקו את התוכנית בלוח ה-UNO. מדדו באמצעות משקף תנודות את האות המתקבל.

שאלה 6.6

נתונה התוכנית שלהלן:

```

1. int val1, val2;
2. int sum;
3. void setup() {
4.     Serial.begin(19200);
5. }
6. void loop() {
7.     Serial.print("Enter first number:");
8.     while (Serial.available()==0){ }
9.     val1 = Serial.parseInt();
10.    Serial.println(val1);
11.    Serial.print("Enter second number:");
12.    while (Serial.available()==0){ }
13.    val2 = Serial.parseInt();
14.    Serial.println(val2);
15.    sum = val1 + val2;
16.    Serial.print("The sum is ");
17.    Serial.println(sum);
18. }
```

- א. הסבירו את ההוראות בשורות 8 ו-9.
- ב. עקבו אחר התוכנית. מהו פלט התוכנית?
- ג. מה התוכנית מבצעת?

שאלה 6.7

כתבו תוכנית הקולטת שני מספרים ממשיים (float) מתוכנת המוניטור ומציגה את מכפלתם. הנחיה: כדי לקלוט מספר ממשי יש פעולה שאינה מופיעה בטבלה 6.1. חפשו באינטרנט פעולה במחלקה Serial שמאפשרת זאת. היעזרו גם בדוגמה 2.

שאלה 6.8

כתבו תוכנית הקולטת מחרוזת תווים שאורכה 15 תווים לכל היותר. אם אורך המחרוזת שנקלטה קטן מ-10 תווים, הציגו את ההודעה "too small". אחרת – אם אורך המחרוזת אינו קטן מ-10 – הציגו את התווים במקומות הזוגיים בלבד. הערה: מחרוזת בשפה זו היא בעצם מערך של תווים.

שאלה 6.9

כתבו תוכנית הקולטת מחרוזת תווים שאורכה לכל היותר 25 תווים ומסתיימת בתו '#'.
התוכנית תציג את המחרוזת שנקלטה.

בדקו את התוכנית עבור דוגמאות הקלט האלה:

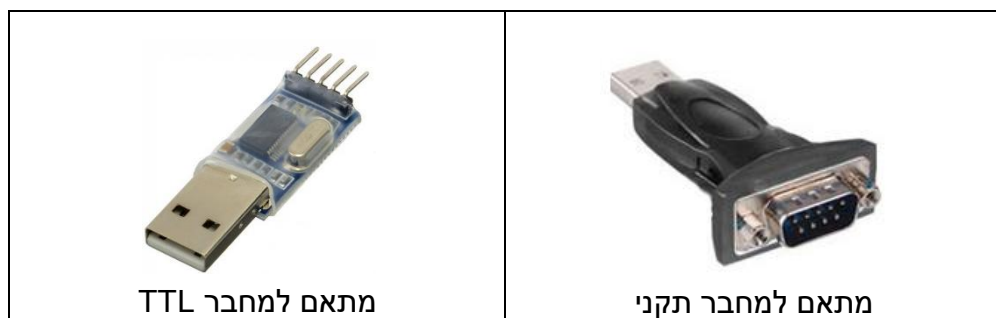
1. Hello#
2. Hello, #world

האם הפתרון שהוצע פועל באופן תקין על שתי הדוגמאות? אם לא, הציעו פתרון.

הערה: מחרוזת בשפה זו היא בעצם מערך של תווים.

6.3 תקשורת טורית במחשב האישי

כאמור, רוב המחשבים כיום אינם מצוידים בהתקן תקשורת טורית. יש מתאמים מתקן USB לתקן RS232.



כאשר עובדים עם לוח Arduino אין צורך במתאם, מכיוון שהוא מובנה בלוח.

כאשר משתמשים במתאם של הלוח למטרת צריבה או תקשורת מול תוכנת ה-Monitor אין אפשרות להשתמש במתאם זה, אלא רק במתאם חיצוני. לכן, כאשר נשתמש במתאם שעל הלוח נסגור את תוכנת המוניטור כדי לאפשר לתוכנה אחרת להשתמש בו.

6.4 תקשורת טורית בשפת C#

בשפת C# יש מחלקה בשם `SerialPort`, הנמצאת במרחב השמות `System.IO.Ports`. כדי להשתמש במחלקה זו נוסיף לתוכנית שלנו את מרחבי השמות האלה:

```
using System.IO.Ports;
```

המחלקה `SerialPort`

בטבלה 6.2 מוצגים חלק ממאפייני המחלקה

שם	תיאור
BaudRate	קצב ערוץ התקשורת הטורית
Databits	מספר הסיביות ביחידת מידע
Parity	סיבית ביקורת
PortName	שם מפתח התקשורת
StopBits	מספר סיביות סיום (עצירה)
ReadTimeout	זמן קצוב לקריאה
WriteTimeout	זמן קצוב לכתיבה
IsOpen	ערך המציין אם הערוץ פתוח

טבלה 6.2 חלק ממאפייני המחלקה `SerialPort`

בטבלה 6.3 מוצג ממשק חלקי של פעולות המחלקה

שם הפעולה	תיאור
<code>SerialPort()</code>	פעולת בנייה – מחדל
<code>SerialPort(String, Int32, Parity, Int32, StopBits)</code>	פעולת בנייה המקבלת את שם המפתח הטורי, קצב, סיבית ביקורת, מספר סיביות מידע וסיביות סיום.
<code>void Open()</code>	פותחת את ערוץ התקשורת. קובע <code>IsOpen = false</code>
<code>void Close()</code>	סגירת ערוץ התקשורת. קובע <code>IsOpen = false</code>
<code>string ReadLine()</code>	קורא מחרוזת מחוצץ הקלט עד לסימן שורה חדשה <code><\n></code> .
<code>void WriteLine(string text)</code>	כותב את המחרוזת <code>text</code> אל חוצץ הפלט.

טבלה 6.3 ממשק חלקי של פעולות המחלקה `SerialPort`

דוגמה 6.6: הגדרת מפתח טורי במחשב האישי בשפת C#

```

1. using System.IO.Ports;
2. class Program
3. {
4.     static SerialPort serial;
5.     static void Main(string[] args)
6.     {
7.         serial = new SerialPort("COM4",19200,Parity.None,8,StopBits.One);
8.         serial.ReadTimeout = 1000;
9.         serial.WriteTimeout = 1000;
10.        serial.Open();
11.        // ToDo
12.    }
13.}

```

הסבר התוכנית

שורה 4: הגדרת עצם מהמחלקה SerialPort

שורה 7: בנייה של עצם עם תקשורת בפרמטרים האלה:

PortName = "COM4"	כשם המפתח שאליו הלוח מחובר
BauRate = 19200	
Parity = Parity.None	ללא סיבית ביקורת
DataBits = 8	
StopBits = StopBits.One	סיבית סיום אחת

שורה 8: קביעת זמן קצוב של שנייה אחת. אם כעבור פרק זמן זה אין קלט, הפעולה מסתיימת.

שורה 9: קביעת זמן קצוב של שנייה אחת. אם כעבור פרק זמן זה אי-אפשר להשלים את הפלט, הפעולה מסתיימת.

שורה 10: פתיחת ערוץ התקשורת לשידור וקליטה. **אם ערוץ התקשורת תפוס על ידי תוכנת המוניטור, לא מתאפשרת פתיחה של הערוץ!**

דוגמה 6.7: הפעלת נורת לד בלוח UNO באמצעות פקודה מתוכנה ב-C#

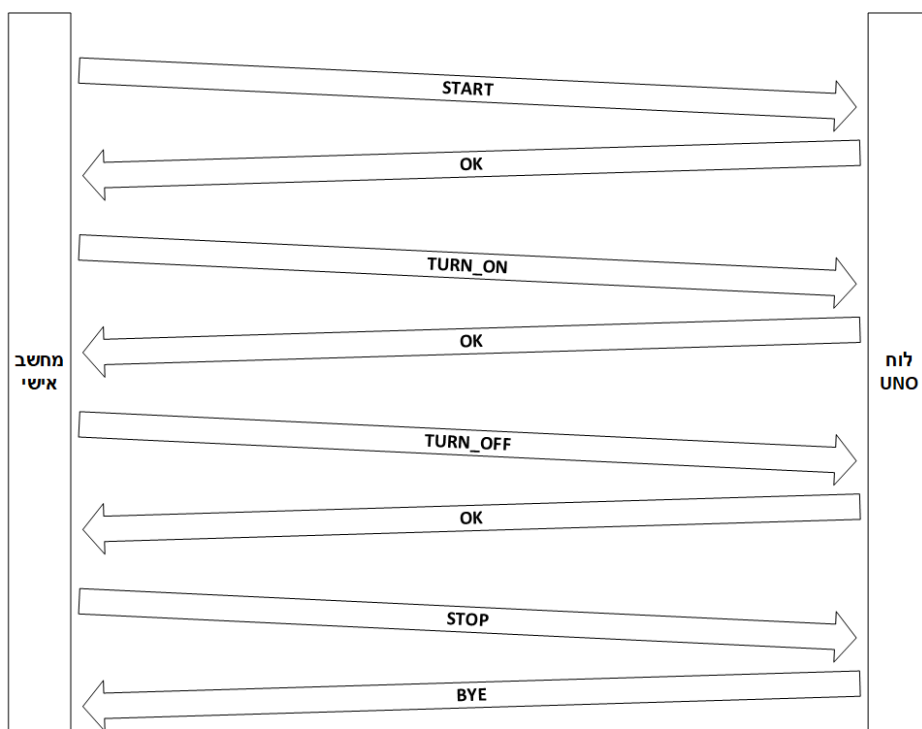
בדוגמה זו נכתוב שתי תוכניות: אחת ללוח UNO והשנייה למחשב אישי בשפת C#. התקשורת בין הלוח למחשב תהיה טורית בקצב 19200.

מטרת התוכנית במחשב האישי היא להציע למשתמש את אפשרויות הפעולה האלה:

1. התחברות ללוח UNO
2. הדלקת נורת לד
3. כיבוי נורת לד
4. התנתקות מלוח ה-UNO וסגירת התוכנית.

מטרת התוכנית בלוח ה-UNO היא להמתין לקבלת הוראות מהמחשב האישי ולמלא אותן. לאחר מילוי הוראה תישלח הודעה חוזרת ללוח, המאשרת את מילוי ההוראה.

באיור 6.4 מוצג פרוטוקול ההעברה בין המחשב ללוח



איור 6.4 פרוטוקול ההעברה בין המחשב ללוח

צד לוח ה-UNO

לוח ה-UNO נמצא במצב המתנה לקבלת ההודעה START. עם קבלת הודעה זו, התוכנה עוברת למצב שבו היא מצפה לקבל אחת משלוש הודעות: TURNON, TURNOFF או STOP. לאחר המעבר נשלחת הודעת אישור OK, למחשב האישי.

עם קבלת הודעה, מתבצע פענוח שלה:

הודעת TURNON – התוכנה מדליקה את נורת הLED ושולחת הודעת אישור OK.

הודעת TURNOFF – התוכנה מכבה את נורת הLED ושולחת הודעת אישור OK.

הודעת STOP – התוכנה חוזרת למצב המתנה לקבלת הודעת START ושולחת אישור BYE.

```
// הגדרת קבועים
#define ARRAYSIZE 25
#define START "START"
#define STOP "STOP"
#define TURNON "TURNON"
#define TURNOFF "TURNOFF"
#define OK "OK\n"
#define BYE "BYE\n"

// הגדרת משתני התוכנית
char val[ARRAYSIZE]; // מערך תווים לקליטת מחרוזת הודעה
int s; // אורך המחרוזת שנקלטה
bool isWaiting; // משתנה להגדרת מצב המתנה לתקשורת
int ledPin = 13; // הדק הLED

void setup() {
  Serial.begin(19200); // תיחול תקשורת טורית
  isWaiting = true; // אתחול למצב המתנה
  pinMode(ledPin, OUTPUT); // הגדרת הדק ספרתי כפלט
  digitalWrite(ledPin, LOW); // כיבוי הLED
}
```

```

void loop() {
  while (Serial.available() > 0) {           // המתנה לקבלת מידע
    s = Serial.readBytesUntil('#', val, ARRAYSIZE); // # קליטה עד קבלת תו
    val[s] = 0;                             // הוספת ערך 0 – סוף מחרוזת
    if (isWaiting && strcmp(val, START) == 0) { // המתנה לקבלת הודעת התחברות
      isWaiting = false;                     // מעבר למצב הוראות
      Serial.print(OK);                      // אישור מעבר לקבלת הוראות
    }
    else if (!isWaiting) {
      if (strcmp(val, STOP) == 0) { // קבלת הודעת ניתוק
        isWaiting = true;
        Serial.print(BYE); // שליחת אישור ניתוק
      }
      else if (strcmp(val, TURNON) == 0) { // קבלת הודעת הדלקה
        digitalWrite(ledPin, HIGH);
        Serial.print(OK);
      }
      else if (strcmp(val, TURNOFF) == 0) { // קבלת הודעת כיבוי
        digitalWrite(ledPin, LOW);
        Serial.print(OK);
      }
    }
  }
}

```


צד המחשב האישי

```

using System;
using System.IO.Ports;      // מרחב שמות של מפתח תקשורת
namespace Serial2
{
    class Program
    {
        static SerialPort serial; // עצם מסוג תקשורת טורית
        static bool keepContinue; // משתנה בוליאני לביצוע הלולאה הראשית
        static int choice;        // משתנה עבור הבחירה של אפשרויות התפריט
        static bool isConnected;  // משתנה בוליאני לציין מצב התחברות ראשוני
        static void Main(string[] args)
        {
            isConnected = false; // קביעת מצב לא מחובר
            // פעולת בנייה לעצם של תקשורת טורית עם אתחול הפרמטרים
            serial = new SerialPort("COM4", 19200, Parity.None, 8, StopBits.One);
            serial.ReadTimeout = 1000; // קוצב זמן לביצוע קריאה
            serial.Open(); // פתיחת ערוץ התקשורת
            keepContinue = true; // אתחול משתנה הלולאה
            // הצגת תפריט עבור המשתמש
            Console.WriteLine("\nLED CONTROL UNO");
            Console.WriteLine("1. CONNECT");
            Console.WriteLine("2. TURN ON");
            Console.WriteLine("3. TURN OFF");
            Console.WriteLine("4. DISCONNECT");
            Console.Write("Enter your choice <1..4> ");
            while (keepContinue)
            {
                choice = Console.ReadKey().KeyChar; // קליטת תו
                switch (choice) // משפט ברירה רב-מצבי
                {
                    case '1': // התחברות ראשונית
                        if (!isConnected) // אם עדיין לא מחובר
                        {
                            serial.WriteLine("START#"); // שליחת הודעת התחברות
                        }
                        break;
                }
            }
        }
    }
}

```

```

case '2':      // דרישה להדלקת הLED
    serial.WriteLine("TURNON#");    // שליחת הודעת הדלקה
    break;
case '3':      // דרישה לכיבוי הLED
    serial.WriteLine("TURNOFF#");    // שליחת הודעת כיבוי
    break;
case '4':      // דרישה להתנתקות
    serial.WriteLine("STOP#");        // שליחת הודעת התנתקות
    break;
}
try // ניסיון לביצוע הקריאה בקטע הבא
{
    string message = serial.ReadLine(); // קלט מחרוזת
    if (message == "BYE") // אישור התנתקות
    {
        Console.WriteLine("Good Bye\nPress any key...");
        isConnected = false; // עדכון משתנים בוליאניים לסיום
        keepContinue = false;
    }
    else if (message == "OK") // קבלת אישור להודעות
    {
        switch (choice)
        {
            case '1': // אישור התחברות
                Console.WriteLine("\nYou are CONNECTED");
                isConnected = true;
                break;
            case '2': // אישור הפעלה
                Console.WriteLine("\nLED IS ON");
                break;
            case '3': // אישור כיבוי
                Console.WriteLine("\nLED IS OFF");
                break;

```

```

    } // הצגה חוזרת של התפריט
    Console.WriteLine("\nLED CONTROL UNO");
    Console.WriteLine("1. CONNECT");
    Console.WriteLine("2. TURN ON");
    Console.WriteLine("3. TURN OFF");
    Console.WriteLine("4. DISCONNECT");
    Console.WriteLine("Enter your choice <1..4> ");

    }
}
catch (TimeoutException) { } // אם ניסיון הקלט לא הצליח
}
serial.Close(); // סגירת מפתח התקשורת בסיום התוכנית
}
}
}

```

 תרגול

שאלה 6.10

כתבו תוכנית ללוח UNO שמפעילה נורת לד המחוברת ליציאה אנלוגית 11, באמצעות ערך מספרי שיתקבל בערוץ התקשורת הטורית באמצעות ה-Serial Monitor. הערך המספרי שיישלח הוא בין 0 ל-255.

שאלה 6.11

כתבו תוכנית ללוח UNO שמפעילה נורת לד המחוברת ליציאה אנלוגית 11, באמצעות ערך מספרי שיתקבל בערוץ התקשורת הטורית באמצעות תוכנה בשפת C# שנכתבה עבור המחשב האישי. הערך המספרי שיישלח הוא בין 0 ל-255.

בצד המחשב האישי ייקלט ערך מספרי מהמשתמש ולאחר מכן יישלח בערוץ התקשורת הטורית אל לוח ה-UNO.

בצד הלוח תהיה המתנה לקבלת ערך מספרי. לאחר קבלת הערך תעודכן היציאה האנלוגית.

שאלה 6.12

רוצים להפעיל נוסף על נורת הלד מדוגמה 4 גם זמזם.

- א. ציינו בתרשים התקשורת שבאיור 4 את התוספת הדרושה.
- ב. מהן ההודעות שצריך להוסיף בצד הלוח ובצד המחשב האישי?
- ג. מהן ההוראות שיש להוסיף בצד המחשב כדי לעדכן את התפריט?
- ד. הוסיפו לתוכנת צד המחשב את ההוראות הנדרשות לשידור הבקשות להפעלת הזמזם ואת ההוראות המקבלות את אישור הבקשה.
- ה. הגדירו בצד הלוח את ההדק שאליו יחובר הזמזם. הוסיפו הוראות הקולטות את הבקשה להפעלת צליל ואת ההוראה להפסקת הצליל.
- ו. הפעילו את התוכנה בלוח ולאחר מכן במחשב. ודאו כי תוכנת המוניטור אינה מופעלת וכי עדכנתם את שם מפתח התקשורת על פי מחשבכם (COM4 או אחר).

סיכום פרק 6

בפרק 6 למדתם את הנושאים האלה:

1. תקשורת טורית ותקשורת מקבילית
2. תקן תקשורת טורית RS232
3. קווי התקשורת TX ו-RX
4. מאפיינים חשמליים של קו תקשורת
5. שיטת שידור א-סינכרונית
6. מבנה מסגרת שידור: סביבת התחלה, מידע, סיבית ביקורת וסיביות סיום
7. זמן סיבית וקצב שידור
8. זמן מסגרת וקצב שידור מסגרות
9. פעולות המחלקה Serial
10. פעולות כתיבה
11. פעולות קריאה תווים, מספרים ומחרוזות
12. קריאה וכתיבה באמצעות המוניטור
13. תקשורת טורית במחשב האישי
14. תקשורת טורית בשפת C# והמחלקה SerialPort
15. מאפיינים ופעולות המחלקה SerialPort
16. יצירת פרוטוקול תקשורת בין מערכות
17. יישום פרוטוקול התקשורת בין המחשב ללוח UNO
18. כתיבת תוכנית דוגמה להפעלה וכיבוי של לד מרחוק בתקשורת טורית.