

## תקשורת טורית

### דפי משימות

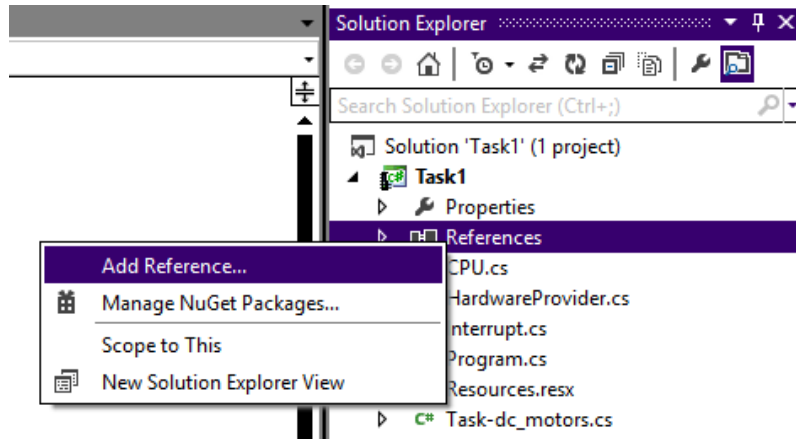
#### ציוד נדרש:

- ערכת פיתוח
- מודול Bluetooth HC06
- טפולון נייד
- אופציות:
  - דוחף זרם ומנועי DC
  - מנוע sevro
  - תצוגת LCD
  - מקלדת 4X4
  - כרטיס זיכרון SD
  - ועוד

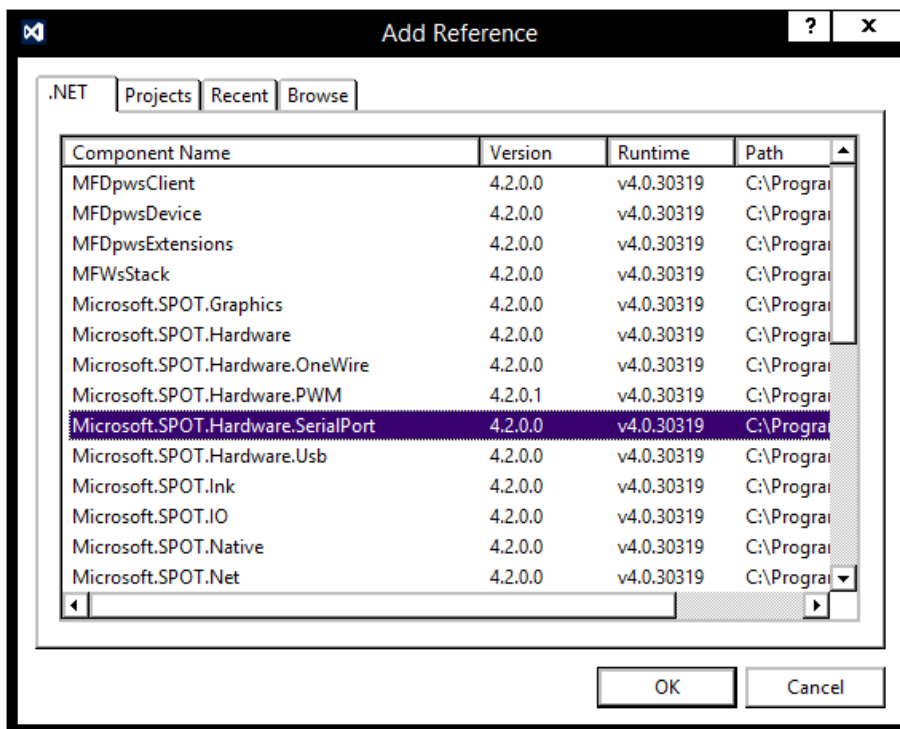
#### מהלך הניסוי

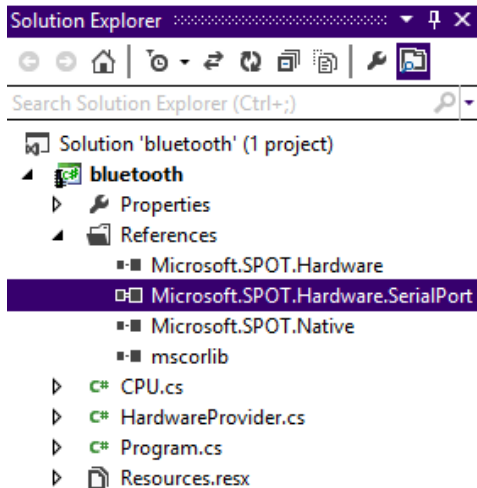


1. בניסוי זה נעשה שימוש בתקשורת טורית. ניתן לתקשר עם רכיבים רבים בצורה זאת. כך למשל: מודול Bluetooth, מודם GSM, כתיבה וקריאה מכרטיסי זיכרון ועוד.
2. נפתח פרויקט חדש לעבודה עם הבקר.
3. בכדי שנוכל לתקשר בתקשורת טורית, נוח להשתמש במחלקה בנויה הייעודית לכך. לשם כך, עלינו להכיר מחלקה זאת לסביבת פיתוח ע"י הכללת הקבצים שלה בתוך הפרויקט. ניתן לעשות זאת באופן הבא:
  - a. ב Visual Studio נלחץ עם המקש הימני של העכבר על References שבחלונית ה Solution Explorer כמתואר באיור:



- b. בתפריט שתפתח נבחר את האפשרות העליונה Add References...
- c. בחלון שיפתח נחבר את בלשונית .Net. בה נגיע לרשומה: Microsoft.SPOT.Hardware.SerialPort, נסמן אותו ונלחץ על הכפתור OK שבתחתית החלונית.





4. בחלונית Solution Explorer נוכל לראות שההרחבה התווספה בהצלחה:

5. נוסיף את ההרחבה גם ל `using` בתוך הקוד שבקובץ `Program.cs` של הפרויקט:

```
using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using STM32;
using System.Text;
using System.Threading;
using System.IO.Ports;
```

6. נגדיר מחוץ לפונקציה הראשית `Main()` את העצם החדש לתקשורת טורית:

```
static SerialPort sp = new SerialPort(SerialPorts.COM2,
                                     9600,
                                     Parity.None,
                                     8,
                                     StopBits.One);
```

7. הסבר מקיף על השדות בתוך ה `constructor` (הפעולה הבונה) ניתן למצוא בנספח שבסוף הפרק.

8. נתחיל מטיפול בשידור מסר ע"י הבקר ולאחר מכן נעבור גם לקליטת מסר והפעלת לד במידה וזה מה שנדרש ממנו במסר שקלט.

9. מחוץ ל פונקציית ה `Main()` נגדיר את הלד אותו נרצה להדליק מאוחר יותר:

```
static OutputPort led = new OutputPort(On_Board.blue_Led, false);
```

10. בתוך ה `Main()` נפתח את הפורט הטורי לעבודה:

```
sp.Open();
```

11. נגדיר מערך בייטים ונכניס אליו את המידע אותו נרצה לשדר. שימו שמחרוזת הטקסט שתשלח חייבת להסתיים ב \r

```
byte[] arr = Encoding.UTF8.GetBytes("Ready for instructions\r");
```

לחילופין, ניתן להגדיר משתנה מטיפוס string ועליו לעשות את ההמרה למערך:

```
string st = "Ready for instructions\r";
byte[] arr = Encoding.UTF8.GetBytes(st);
```

12. נשדר את הטקסט בתקשורת טורית רק בתנאי שנלחץ לחצן כחול

```
if (button.Read()) sp.Write(arr, 0, arr.Length);
```

13. השדה הראשון הוא מערך הבייטים אותו יש לשדר, השני ההזזה בתוכו ממנה יתחיל השידור והשלישי מספר הבייטים שיש לשדר.

14. נסיים את ה Main() עם לולאה אינסופית "שתעטוף" את השידור:

```
while (true)
{
    if (button.Read()) sp.Write(arr, 0, arr.Length);
}
```

15. ניתן לשדרג במקצת את התוכנית ע"י הוספת חיווי ויזואלי בלדים שע"ג הכרטיס.

16. בסה"כ, בשלב זה נקבל את התוכנית הבאה:

```
using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using Microsoft.SPOT.Hardware.STM32;
using System.Threading;
using System.Text;
using System.IO.Ports;

namespace Task1
{
    public class Program
    {
        static SerialPort sp = new SerialPort(SerialPorts.COM2,
            9600,
            Parity.None,
            8,
            StopBits.One);

        static InputPort button = new InputPort(On_Board.blue_Button,
```



- יש לחבר גם את הפין של האדמה מהפלג של RS-232 להדק GND בערכת הפיתוח.
- את הפין VCC של מודול התקשורת יש לחבר ל 3.3V או ע"פ הוראות יצרן אחרות.

20. מאחר ואנו משתמשים בקווי העברת המידע בלבד (Rx, Tx), עדיף לבחור ב COM2 או COM3 בהם אין חפיפה בפנים אלה עם רכיבים נוספים בערכת הפיתוח. בתרגיל זה בחרנו להשתמש ב COM2.

21. מאחר ורגל השידור של מודול ה Bluetooth (או של התכן תקשורת טורית אחר) היינה רגל הקליטה של הבקר וגם רגל הקליטה של מודול ה Bluetooth היינה רגל השידור של הבקר, יש להצליב בין שני החיבורים.

22. אם נחבר מודול Bluetooth דרך COM2, הרי שטבלת החיבורים שלו תהינה:

שם הפין במודול	שם הפין בערכת פיתוח	הערות
VCC	3V	
GND	GND	
TXD	A3	
RXD	A2	

23. ואם החיבור מתבצע דרך COM3, הרי שנקבל:

שם הפין במודול	שם הפין בערכת פיתוח	הערות
VCC	3V	
GND	GND	
TXD	D9	
RXD	D8	

24. נחבר את הערכה למחשב ע"י שני חיבורי ה USB ונצרוב את התוכנה לבקר ע"י לחיצה עם העכבר על בלחצן Start שבסרגל הפקודות.

25. לאחר הצריבה של התוכנית, נפתח תוכנה במחשב בה ניתן לקלוט מסרים מתקשורת טורית ונגדיר את מאפייניה ע"פ מה שרשמנו ב Visual Studio וצרבו בבקר. כלומר:

שם המאפיין	ערך
שם הפורט	COM2
Baud rate	9600
Parity	none
Data bits	8
Stop bits	1

26. אחרי שהתוכנה פתוחה ומוכנה, ניתן ללחוץ על לחצן ה reset של הבקר (הלחצן השחור) בכדי שישדר בשנית את המידע.
27. הכיתוב "Ready for instructions" שרשמנו יופיע בחלון התוכנה.
28. ניתן לממש את התקשורת הטורית דרך Bluetooth ולהתחבר לבקר ממחשב בעל התקן זה או מטלפון נייד באמצעות אפליקציה מתאימה כדוגמת Bluetooth terminal.

## עבודה עם מודול Bluetooth מול טלפון נייד

1. נחבר את המודול לבקר ע"פ העקרונות שנדונו בסעיפים הקודמים. לנוחיותנו נסכמם בטבלה:

שם הרגל במודול	RXD	TXD	GND	VCC	הערות
COM1	A9	A10	GND	3V	יש לנתק את ה microUSB
COM2	A2	A3	GND	3V	
COM3	D8	D9	GND	3V	

2. לאחר חיבורם של כ 4 הדקי המודול לכרטיס הבקר, נחבר את הבקר לחשב ע"י שני חיבורי ה USB.
3. ההסבר שלהלן מתייחס לטלפון נייד Samsung galaxy, אך מתוכו ניתן ללמוד גם לגבי טלפונים ניידים של חברות אחרות.
4. נוריד ונתקין בתוך הטלפון הנייד את האפליקציה Bluetooth terminal המאפשרת לשלוח ולקבל הודעות טקסט דרך תקשורת טורית ב Bluetooth.
5. נפעיל את האפליקציה.
6. במידה וה Bluetooth במכשיר הסלולרי שלכם כבוי, האפליקציה תקפיץ חלון בקשת אישור הפעלת Bluetooth במכשיר שיש לאשרה ע"י לחיצה על כפתור "כן"
7. יפתח חלון בו ניתן להזין טקסט ובחלקו הימני העליון יופיע כיתוב not connected
8. נלחץ על לחצן מגע שמאלי בטלפון הנייד ומתוך התפריט שתפתח נבחר את האפשרות Connect a device – Insecure
9. תפתח חלונית עם רשימת התקני ה Bluetooth המוכרים למכשיר הטלפון הנייד שלכם.
10. במידה והנכם עובדים עם מודול ה Bluetooth בפעם הראשונה, הוא לא יופיע ברשימה.
11. במידה ואתם לא רואים את ההתקן HC-06 (מודול ה Bluetooth אותו חיברנו לערכת הפיתוח) בתוך רשימת ההתקנים המוכרים למכשיר, לחצו על הלחצן Scan for devices שבתחתית הרשימה ולאחר המתנה קצרה הרכיב ייתוסף לרשימה.
12. מודול ה Bluetooth יופיע ברשימה בשם HC-06 וכתובת כגון 98:D3:31:B3:0D:7D
13. נלחץ עליו.
14. בחלקו העליון של החלון תופיע הודעה Connecting ולאחר סיום מוצלח של החיבור: connected: HC-06
15. כעת ניתן ללחוץ על הלחצן הכחול שבערכת הפיתוח ולקבל במכשיר הנייד את ההודעה אותה שולח הבקר: "Ready for instructions".
16. במידה ונרצה לשדר מידע לבקר בתקשורת טורית דרך ה Bluetooth (למרות שבשלב זה עדיין לא כתבנו אל תוך הבקר את התוכנה המעבדת את הטקסט שייקלט), פשוט נקליד את הטקסט (באנגלית) בתוך חלון האפליקציה ונלחץ על הלחצן send.



## קליטת נתונים בבקר

1. בהמשך לתוכנת השידור, נגדיר בתוך ה- `Main()` את הפונקציה שנרצה שתבצע כאשר יתקבל מידע דרך ערוץ התקשורת הטורית

```
sp.DataReceived += new SerialDataReceivedEventHandler(sp_DataReceived);
```

2. כעת ניתן לעבור לכתיבת הפונקציה אליה הפננו מקודם, ה- `sp_DataReceived`. לשם כך מחוץ ל- `Main()` נכתוב:

```
static void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    Thread.Sleep(50);
}
```

3. נגדיר משתנה חדש מטיפוס `int` ונכניס בו את מספר הבייטים שיש לקרואה.

```
int num = sp.BytesToRead;
```

4. נגדיר חוצץ ממערך של בטים אליו נקראה את המידע.

```
byte[] InBuffer = new byte[num];
```

5. נבצעה את הקריאה מאובייקט התקשורת שלנו

```
sp.Read(InBuffer, 0, InBuffer.Length);
```

6. מאחר ו- `InBuffer` מוגדר מחוץ ל- `Main()`, ערכו ישמר בין קליטת נתונים אחת לשנייה, על כן יש למחוק את המידע הקודם ששמור בו לפני שמכניסים אליו את המידע החדש. ניתן לעשות זאת כך:

```
for (int i = 0; i < InBuffer.Length; i++)
{
    InBuffer[i] = 0;
}
```

7. נמיר את המערך של הבייטים שקלטנו למחרוזת `string` אותה נוכל להציג בחלונית `Output` של סביבת הפיתוח

```
char[] arr = Encoding.UTF8.GetChars(InBuffer);
string st = new string(arr);
Debug.Print(st);
```

לחילופין, ניתן לבצע זאת גם כך:

```
foreach (var item in InBuffer)
{
    st += ((char)item).ToString();
}
```

8. נבדוק העם הגיע איפשהו בתוך המידע שהתקבל הטקסט "green". במידה וכן, נדליק את הLED הירוק כמבוקש ונחזיר את ההודעה לשולח על כך שהLED הודלק.

```
if (st.IndexOf("green") != -1)
{
    led.Write(true);
    byte[] outBuf = Encoding.UTF8.GetBytes("The green LED is turned on!\r\n");
    sp.Write(outBuf, 0, outBuf.Length);
}
```

9. גם בקריאת נתונים, ניתן להוסיף LED שלא בשימוש למטרת חיזוי ויזואלי.

10. בסה"כ קיבלנו את התוכנית הבאה:

```
using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using Microsoft.SPOT.Hardware.STM32;
using System.Threading;
using System.Text;
using System.IO.Ports;

namespace Task1
{
    public class Program
    {
        static SerialPort sp = new SerialPort(SerialPorts.COM2,
            9600,
            Parity.None,
            8,
            StopBits.One);
        static InputPort button = new InputPort(On_Board.blue_Button,
            true,
            ResistorModes.PullDown);
        static OutputPort led = new OutputPort(On_Board.blue_Led, false);
        static OutputPort led2 = new OutputPort(On_Board.red_Led, false);
        static OutputPort led3 = new OutputPort(On_Board.green_Led, false);

        public static byte[] InBuffer = new byte[100];

        public static void Main()
        {
            sp.DataReceived += new SerialDataReceivedEventHandler(sp_DataReceived);
```

```

sp.Open();
string st = "Ready for instructions\r";
byte[] arr = Encoding.UTF8.GetBytes(st);
while (true)
{
    if (button.Read())
    {
        sp.Write(arr, 0, arr.Length);
        led.Write(true);
        Thread.Sleep(500);
        led.Write(false);
        Thread.Sleep(500);
    }
}
}

static void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    led2.Write(true);
    Debug.Print("Recieved");
    Thread.Sleep(50);
    for (int i = 0; i < InBuffer.Length; i++)
    {
        InBuffer [i] = 0;
    }
    sp.Read(InBuffer, 0, sp.BytesToRead);

    string st = "";
    foreach (var item in InBuffer)
    {
        st += ((char)item).ToString();
    }
    Debug.Print(st);
    Thread.Sleep(500);
    led2.Write(false);

    if (st.IndexOf("green") != -1)
    {
        led3.Write(true);
        byte[] outBuf = Encoding.UTF8.GetBytes(
            "The green LED is turned on!\r");
        sp.Write(outBuf, 0, outBuf.Length);
    }
}

```

```
}  
}  
  
}  
}
```

11. נצרוב אותה לבקר ע"י לחיצה עם העכבר על בלחצן Start שבסרגל הפקודות.
12. לאחר הצריבה של התוכנית, נשלח בתקשורת טורית מהמחשב או מהטלפון הנייד את הטקסט שנרצה שייקלט ע"י הבקר. אם נשלח את הטקסט "green", הLED הירוק ידלק ולמפעיל תשודר ההודעה "The green LED is turned on!".
13. בהצלחה!

## משימות

### משימה 1: רובוט נשלט דרך טלפון נייד

כתוב תוכנית המפעילה רובוט דרך מכשיר הטלפון הנייד באמצעות השימוש ב Bluetooth. על התוכנית לעמוד בדרישות הבאות:

1. בהדלקה הרובוט משדר למפעיל הודעת טקסט דרך Bluetooth בה ידיע שהוא ממתין לקוד משתמש שיאפשר לו גישה לתפעול הרובוט.
  2. על הקוד גישה להיות מורכב מאותיות וספרות ולהכיל לפחות 5 תווים.
  3. כל קוד גישה תקין מאפשר זמן תפעול שונה של רובוט.
  4. בקליטת הקוד, הרובוט יחזיר הודעה האם הקוד תקין וכמה זמן עומד לרשות המפעיל לתפעול הרובוט.
  5. לאחר ההזדהות, המשתמש יוכל לשלוח לרובוט את ההודעות הבאות:
    - a. forward – סע קדימה
    - b. left – פנה שמאלה
    - c. right – פנה ימינה
    - d. back – סע אחורה
    - e. stop – עצור
    - f. patrol – סייר סביב המיקום העכשווי
- בהעדר דוחף הזרם והגלגלים, כיוון נסיעת הרובוט יכול להיות מסומן ע"י הדלקה של אחד מארבעת הלדים שע"ג הכרטיס, בהתאם לכיוון הנסיעה הרצוי.
6. בקבלת הפקודה, הרובוט יחזיר הודעה מתאימה למפעיל. למשל עבור פקודת הנסיעה קדימה, הרובוט יחזיר: "Going forward".
  7. במקרה של תעונה (שינוי חד פתאומי בתאוצה), הרובוט יעצור וישלח הודעה מתאימה למפעיל.
  8. לחיצה על לחצן חרום (הלחצן הכחול ע"ג הכרטיס) תגרום לעצירה מיידים ושליחת הודעה מתאימה למפעיל.
  9. אחת לדקה הרובוט ישלח הודעה למפעיל בה ידיעה כמה זמן נותר לתפעול.
  10. בסיום זמן התפעול, הרובוט יעצור ויבקר מהזדהות בשנית. במידה ונגמר הזמן לקוד מסוים, תתקבל הודעה מתאימה.
- הערה: ניתן לשלב במשימה זאת גם:
- חיישן מרחק אחד או יותר בכדי למנוע תאונות ולעקוף מכשולים תוך שמירה על כיוון תנוע רצוי.
  - תצוגת LCD להצגת הודעות.

- תאורה קדמית ואחורית המופעלת כאשר הרובוט נוסע בכיוון זה.
- במידה ומשתמשים במנועים:
  - תחילת תנוע ועצירה הדרגתיים.
  - הפעלת אורות משטרה (לדים כחול-אדום) ע"פ דרישת המפעיל.

ניתן להתרשם מרובוט דומה בקישור הבא: <https://www.youtube.com/watch?v=nlWxjXyERMM>  
מעוניינים יכולים להתרשם מסרטונים נוספים של רובוטים שונים בכדי לקבל רעיונות לשדרוגים...

## משימה 2: שלט רחוק למכשירים ביתיים

כתוב תוכנית שתאפשר הפעלת צרכנים ביתיים (כגון בוילר, תאורה, חימום, פתיחת/סגירת התריסים וכד') ע"י שליחת הודעה מתאימה דרך Bluetooth לבקר. על התוכנה לעמוד בדרישות הבאות:



1. לאפשר הפעלה / כיבוי מיידים של 4 צרכנים ביתיים (4 הלדים שע"ג ערכת הפיתוח) ע"י שליחת הודעה כגון: "salon lights .on"
2. לאפשר הדלקה / כיבוי של אותם הצרכנים מרגע זה למשך פרק זמן מוגדר ע"י שליחת הודעה כגון: "boiler on for 25 min" או "Heater on for 2h 30min"
3. לאפשר הדלקה / כיבוי של אותם הצרכנים בשעות מוגדרות מראש ע"י שליחת הודעה כגון: "Tris up from 6:00 to 18:30"
4. לאפשר כיבוי מיידים של כל המכשירים ע"י שליחת הודעה מפורשת כגון "all off".
5. כיבוי מיידים של כל המכשירים יתבצע גם ע"י לחיצה על לחצן חרום (לחצן כחול ע"ג הערכה).
6. אחת לדקה הבקר ידווח למחשב את השעה הנוכחית שלו ואת רשימת המכשירים הדלוקים.
7. על כל הוראה שתתקבל בבקר מאת המפעיל, תשלח הודעת תגובה המאפשרת שההוראה נקלטה תקין.
8. בהתחברות ראשונית לבקר יש להזדהות ע"י סיסמא.
9. במידה וסיסמת ההזדהות לא נשלחה תקין 3 פעמים ברציפות, לא תאפשר הזנת סיסמא בפעם נוספת למשך 15 דקות.
10. הבקר יחשב את צריכת ההספק של כל אחד מהמכשירים ואחת לחודש ידווח למחשב ולמפעיל מה הייתה הצריכה החודשים של כל מכשיר בקו"ש ומהי העלות של צריכה זו.

### משימה 3: מעמן כושר אישי

כתבו תוכנה שתנהל את הפעילות הספורטיבית אותה מבצע הנוסע את המכשיר עליו. על התוכנה לעמוד במאפיינים הבאים:

1. התחלת האימון תאופיין ע"י הזזת המכשיר למשך יותר מ 10 שניות אן לחיצה על הלחצן הכחול.
2. סיום האימון מאופיין ע"י מנוחת המכשיר למשך יותר מ 20 שניות או לחיצה על הלחצן הכחול.
3. בסיום האימון המכשיר ידווח דרך הודעה ב Bluetooth את הנתונים הבאים:
  - a. מספר צעדים מתחילת האימון ועד סופו.
  - b. קק"ל אותם שרף המתאמן.
  - c. רמת מאמץ ממוצעת של העימון.
  - d. רמת מאמץ שיא של העימון.
  - e. היסטוריה של הנתונים הנ"ל בשבוע האחרון.
  - f. ממוצע של הנתונים הנ"ל בחודש האחרון.

בהצלחה!



## נספח: יצירת פורט טורי (serial port)

פורט סריאלי הוא פורט מאוד שימושי. הוא מאפשר תקשורת ברמות TTL בפרוטוקול RS232 (פרוטוקול זה היה בשימוש לפני עידן ה-USB, במחשבים ישנים עדיין ניתן למצוא כניסת COM שמיועדת במיוחד לתקשורת מסוג זה. הפרוטוקול עדיין בשימוש כיום, בעיקר ברמות חומרה. לפני שימוש בפורט זה יש לקרוא ולהבין כיצד הוא עובד! כדי ליצור פורט סריאלי נשתמש בעצם **.SerialPort**

לדוגמא:

```
SerialPort sp = new SerialPort(SerialPorts.COM1, 9600, Parity.None,
                                8, StopBits.One);
```

### תכונות:

התכונה	טיפוס	מטרה
BaudRate	int	קובעת את קצב העברת הנתונים.
BytesToRead	int	מונה את מספר הבתים הנמצאים בבאפר הכניסה.
DataBits	int	מגדירה את כמות הביטים המועברים ב-"מילה" אחת.
Handshake	enum Handshake	מגדירה את סוג ה-handshake של הפרוטוקול.
IsOpen	bool	מכילה true אם כן, false אם לא.
Parity	enum Parity	מגדירה האם בפרוטוקול תבצע בדיקת זוגיות או לא, ואם כן מאיזה סוג.
PortName	string	שומרת את שם הפורט.
ReadTimeout	int	קובעת במילי-שניות את כמות הזמן שהפורט ימתין לקלט לפני שישלח Exception שיבשר על על ReadTimeOut. ניתן לקבוע כאין סופי.

מגדירה את כמות ביטי העצירה בפרוטוקול.	enum StopBits	StopBits
קובעת במילי-שניות את כמות הזמן שהפורט ימתין בניסיון לשלוח מידע לפני שישלח Exception שיבשר על WriteTimeOut. ניתן לקבוע כאין סופי.	int	WriteTimeout

### פעולות:

טענת יציאה	טענת כניסה	הפעולה
הפעולה בונה עצם חדש מסוג SerialPort ומאתחלת את הפורט למצב המבוקש.	הפעולה מקבלת שם פורט, קצב העברת מידע, הגדרת זוגיות, מספר ביטי מידע והגדרת ביטי עצירה.	SerialPort (String portName, int baudRate, Parity parity, int (dataBits, StopBits stopBits
הפעולה פותחת את הפורט. ללא ביצוע פעולה זו ניסיון שליחת מידע יוביל לשגיאה!		Open()
הפעולה סוגרת הפורט.		Close()
הפעולה קוראת מהפורט הסריאלי את המידע ומשימה אותו במערך. הפעולה מחזירה את מספר הבתים שנקראו.	הפעולה מקבלת מערך של בתים, אינדקס המציין החל מאיזה תא במערך להזין נתונים וכמות הבתים שיש לקרוא.	int Read (byte[] buffer, int (offset, int count
הפעולה כותבת את המידע המבוקש לפורט הסריאלי. הפעולה מחזירה את מספר	הפעולה מקבלת מערך של בתים, אינדקס המציין החל מאיזה תא	int Write (byte[] buffer, int (offset, int count

הבתים שנכתבו.	במערך לשלוח וכמות הבתים שיש לכתוב.	
הפעולה מרוקנת את תוכן באפר הקלט.		void DiscardInBuffer()
הפעולה מרוקנת את תוכן באפר הפלט.		void DiscardOutBuffer()
שולחת כל מידע שנשאר בבאפר הפלט ומרוקנת אותו.		void Flush()

### מנהלי אירועים:

מטרה	סוג	מנהל האירוע
משוגר כאשר מידע מתקבל בפורט הסריאלי.	SerialDataReceivedEventHandler	DataReceived
משוגר כאשר מתקבלת שגיאה בפורט הסריאלי.	SerialErrorReceivedEventHandler	ErrorReceived