

עבודה עם פסיקות

רקע עיוני

- פעולה בו זמנית של כמה מנגנונים
- באמצע הנסיע הרובוט נתקל במכשול
- באמצע ההיבהוב של הנוריות מתקבלת בקשה אחרת
- מיסוך פסיקות
- העדפת פסיקות

מהלך הניסוי

1. ראשית נפתח פרוייקט חדש לעבודה עם הבקר, כפי שלמדנו לעשות זאת בשיעורים הקודמים.
2. תחילה נגדיר פורט חדש שישמש אותנו לקבלת פסיקה חיצונית לבקר. נעשה זאת מחוץ לפונקציית Main(). לשם הפשטות, במשימה הראשונה נשתמש בלחצן כחול הממוקם ע"ג כרטיס הבקר שאיתו כבר עבדנו. נוסף על כך, יהיה עלינו להגדיר האם ברצוננו להשתמש במסנן המונע ריטוטי מעבר בלחיצת המפסק וכן האם יש צורך בנגדי צימוד (תכונות אלה קיימות גם בהדקי מבוא רגלים הלא משמשים כמבואות פסיקה). התכונה האחרונה אותה יש להגדיר עבור מבוא הפסיקה היא הזווית או הרמה הלוגית בה תתבצע הפסיקה (ב '0' לוגי, '1' לוגי, שתהם, בעליה או בירידה).

```
static InterruptPort pb = new InterruptPort(On_Board.blue_Button,
                                           true,
                                           Port.ResistorMode.PullDown,
                                           InterruptModes.InterruptEdgeHigh);
```

3. כעת, עלינו להגדיר איזו פונקציה ברצוננו שתתבצע כאשר תתקבלנה הפסיקה. ניתן להקיש במקלדת רק את התחלת הפקודה: pb.OnInterrupt += ולהקיש פעמיים על המקש TAB שבמקלדת. Visual Studio ישלים את התוכנית הבאה:

```
static InterruptPort pb = new InterruptPort(On_Board.blue_Button,
                                           true,
                                           Port.ResistorMode.PullDown,
                                           InterruptModes.InterruptEdgeHigh);

public static void Main()
{
    pb.OnInterrupt += pb_OnInterrupt;
```

```

}

static void pb_OnInterrupt(uint data1, uint data2, DateTime time)
{
    throw new NotImplementedException();
}

```

4. עלינו רק להחליף את הקטע המודגש בקוד שאנו רוצים שיתבצע בפסיקה. נניח שמדובר במערכת אזעקה במוזיאון שמדליקה לד ירוק כאשר היא פועלת. כאשר המערכת תזהה פריצה (המסומנת ע"י לחיצה על לחצן כחול) היא תהבהב באור אדום.
5. ניישם את התוכנית באופן הבא:

```

using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using STM32;
using System.Threading;
using System.Text;

namespace Task
{
    public class Program
    {
        static OutputPort led_r = new OutputPort(NoaUpBasic.red_Led, false);
        static OutputPort led_g = new OutputPort(NoaUpBasic.green_Led, false);
        static InterruptPort pb = new InterruptPort(NoaUpBasic.pb2,
                                                    true,
                                                    Port.ResistorMode.Disabled,
                                                    InterruptModes.InterruptEdgeHigh);

        public static void Main()
        {
            pb.OnInterrupt += sensor_OnInterrupt;

            led_g.Write(true);
            Thread.Sleep(-1);
        }

        static void sensor_OnInterrupt(uint data1, uint data2, DateTime time)
        {

```

```

while (true)
{
    led_r.Write(true);
    Thread.Sleep(200);
    led_r.Write(false);
    Thread.Sleep(200);
}
}
}
}
}

```

6. ניתן לשדרג את הניסוי וליצור פסיקות מרובות, לקבוע תנאי סיומם וביצוע מקבילי.

יצירת פורט פסיקה

פורט כניסה דיגיטלי המשגר אירוע לפי סוג הפסיקה המבוקש. כדי להשתמש בפורט מסוים כפורט פסיקה, נשתמש במחלקה **InterruptPort**.

לדוגמא:

```

InterruptPort p = new InterruptPort(Pins.GPIO_PIN_A_1, false,
    Port.ResistorMode.Disabled, Port.InterruptMode.InterruptEdgeHigh);

```

תכונות:

מטרה	טיפוס	התכונה
קוד הזיהוי של הפורט	Cpu.Pin	id
לקבוע האם יבוצע פילטר "גליצים" בפורט.	bool	glitchFilter
מתאר את סוג הפסיקה שהפורט מוגדר לפעול על-פיה.	InterruptMode	interrupt
מאפשר להגדיר פול-אפ או פול-דאון לפורט.	ResistorMode	resistor

פעולות:

טענת כניסה	טענת יציאה	הפעולה
הפעולה מקבלת שם פורט (Cpu.Pin) ערך בוליאני עבור האם להפעיל פילטר גליצ'ים, מצב נגד ומצב פסיקה	הפעולה בונה עצם חדש InterruptPort ומאתחלת את הפורט למצב המבוקש.	OutputPort(Cpu.Pin portId, bool glitchFilter, Port.ResistorMode resistor, Port.InterruptMode interrupt)
	הפעולה מחזירה את הערך הכתוב כרגע בפורט.	bool Read()
	הפעולה מסיימת את הפסיקה הנוכחית ומאפשרת את קריאת הפסיקה הבאה.	void ClearInterrupt()
	הפעולה מבטלת את שיגור האירועים בפסיקות.	void DisableInterrupt()
	הפעולה מפעילה את שיגור האירועים בפסיקות.	* void EnableInterrupt()

*בסוג הפסיקות InterruptEdgeLevelHigh או InterruptEdgeLevelLow האירוע יופעל רק פעם ראשונה בעת פסיקה. כדי להפעיל מחדש את הפסיקה יש להשתמש ב-EnableInterrupt().

מנהלי אירועים:

מנהל האירוע	סוג	מטרה
OnInterrupt	NativeEventHandler	משוגר כשיש פסיקה מהסוג המבוקש בפורט.