

# BEÁGYAZOTT RENDSZEREK ALAPJAI

---

Egyszerű modulok

# Tartalom

- Feladatok egyszerű modulokkal
  - LED
  - Nyomógomb
- Időzítés alapjai
  - System.Threading.Timer osztály
  - Gadgeteer.Timer osztály

# 1. Feladat

Készítsen olyan konfigurációt, mely 2 LED-et és 2 nyomógombot tartalmaz. Ha a jobboldali gombot nyomjuk meg, akkor a jobboldali LED-et kell felgyújtani kék színnel, a baloldali gomb megnyomásával ugyanez történjen a baloldali LED-del. A két gomb egyidejű nyomva tartása esetén mindkét LED kerüljön felgyújtásra piros színnel. A gombok felengedésének hatására a LED-ek nem világíthatnak (tehát csak a gombok lenyomásának hatására villanhatnak fel a LED-ek).

*Hasznos mezők, eseménykezelők:*

```
button.IsPressed
```

```
public delegate void ButtonEventHandler(Button sender, Button.ButtonState state);
```

```
public enum ButtonState  
{  
    Pressed = 0,  
    Released = 1,  
}
```

## 2. Feladat

Készítsen olyan alkalmazást, mely ún. combo-k kezelését teszi lehetővé 2 nyomógomb felhasználásával. A kombinációk hatására a LED-et egy bizonyos színnel kell felgyújtani, mely a kombinációktól függ.

A program a következő kombinációkat kezelje:

- JJBB: zöld
- BJJJ: piros
- BBEE: kék
- EEJJ: fehér

(J: jobboldali gomb, B: baloldali gomb, E: együttes megnyomás)

# System.Threading.Timer

- Lehetővé teszi egy metódus adott idő intervallumonkénti hívását

```
public sealed class Timer: MarshalByRefObject, IDisposable
{
    public Timer(TimerCallback callback, object state, int dueTime, int period);
    public Timer(TimerCallback callback, object state, TimeSpan dueTime, TimeSpan period);

    public bool Change(int dueTime, int period);
    public bool Change(TimeSpan dueTime, TimeSpan period);
    public void Dispose();
}
```

- Konstruktor paraméterei:
  - callback: egy TimerCallback delegate, mely a periodikusan hívni kívánt metódust reprezentálja
  - state: tetszőleges objektum, mely felhasználható a callback metódus hívásakor (lehet null)
  - dueTime: az az idő intervallum, amennyit a Timer indítása előtt kivár a rendszer [ms], 0 ha azonnali indítást szeretnénk
  - period: a metódushívások közti időintervallum [ms]
- A Change metódus segítségével az idővel kapcsolatos paraméterek a példányosítást követően is megváltoztathatók

# System.Threading.Timer

- Rövid példa:

```
public partial class Program
{
    private Timer timer;

    void ProgramStarted()
    {
        Debug.Print("Program Started");
        timer = new Timer(timerCallback, null, 0, 3000);
    }

    private void timerCallback(object state)
    {
        led.BlinkOnce(GT.Color.Red);
    }
}
```

Figyeljük meg a TimerCallback delegate szignatúráját!

```
public delegate void TimerCallback(object state);
```

# Gadgeteer.Timer

- A Microsoft.SPOT.DispatcherTimer egyszerűsített változata
- Egyszerűen újraindítható, indítható, leállítható

```
public void Restart();  
public void Start();  
public void Stop();
```

- Viselkedéstípusokat definiál (BehaviorType)
  - RunOnce: elindul, meghívja a Tick eseményt a megadott idő elteltével (interval), majd leáll („felhúzható óra”)
  - RunContinuously: folyamatosan hívja a Tick eseményt a megadott idő szerint (interval), periodikus üzemmód

```
public enum BehaviorType  
{  
    RunOnce = 0,  
    RunContinuously = 1,  
}
```

- Mindkét viselkedéstípusnál a Tick esemény hívódik meg:

```
public event Timer.TickEventHandler Tick;
```

# Gadgeteer.Timer

```
public class Timer
{
    public Timer(int intervalMilliseconds);
    public Timer(TimeSpan interval);
    public Timer(int intervalMilliseconds, Timer.BehaviorType behavior);
    public Timer(TimeSpan interval, Timer.BehaviorType behavior);

    public Timer.BehaviorType Behavior { get; set; }
    public TimeSpan Interval { get; set; }
    public bool IsRunning { get; }

    public event Timer.TickEventHandler Tick;

    public override int GetHashCode();
    public static TimeSpan GetMachineTime();
    public void Restart();
    public void Start();
    public void Stop();

    public enum BehaviorType
    {
        RunOnce = 0,
        RunContinuously = 1,
    }

    public delegate void TickEventHandler(Timer timer);
}
```

# Gadgeteer.Timer

- Rövid példa:

```
public partial class Program
{
    private GT.Timer timer;

    void ProgramStarted()
    {
        Debug.Print("Program Started");
        timer = new GT.Timer(3000);
        timer.Tick += new GT.Timer.TickEventHandler(timer_Tick);
        timer.Start();
    }

    void timer_Tick(GT.Timer timer)
    {
        led.BlinkOnce(GT.Color.Red);
    }
}
```

Figyeljük meg a TickEventHandler delegate szignatúráját!

```
public delegate void TickEventHandler(Timer timer);
```