

# BEÁGYAZOTT RENDSZEREK ALAPJAI

---

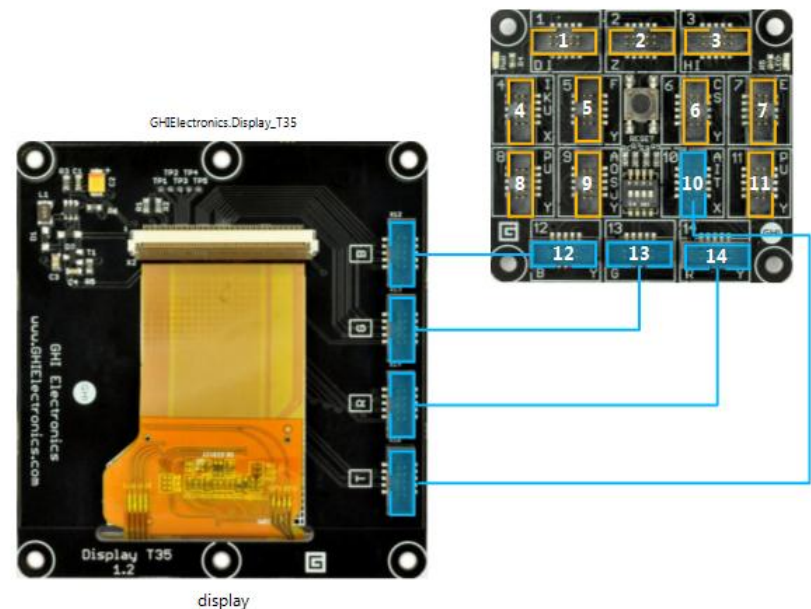
Grafika

# Tartalom

- Grafikus kijelző (T35 Display)
- Rajzolás alapjai
- Grafikus stopper alkalmazás

# Grafikus LCD kijelző (T35 display)

- 320x240 pixeles felbontás
- Érintőképernyőként működik
- Aljzatok: R, G, B, T
  - R, G, B: additív színkeveréshez
  - T (touch): érintés jelzése
- A demópanelen a megjelölt R, G, B, illetve T aljzathoz kell csatlakoztatni!
- A kijelző T kimenetét mindig csatlakoztassuk, különben `NullReferenceException`-t kapunk!



# Display\_T35 osztály

- A Display\_T35 osztály reprezentálja a grafikus kijelzőt
- Alapvető tulajdonságok:
  - Width: a megjelenítő szélessége pixelben
  - Height: a megjelenítő magassága pixelben
  - SimpleGraphics: alapvető rajzolási műveleteket, pl. szöveg, egyszerű alakzatok, képek rajzolását biztosító SimpleGraphicsInterface implementáció.
  - WPFWindow: egy WPF ablakot reprezentáló Window implementáció komplex megjelenítési feladatokhoz. A rajzolási műveleteken kívül lehetővé teszi .NET MicroFramework vezérlők (StackPanel, Canvas, stb.) beágyazását, elhelyezését (layout).

# SimpleGraphics – Egyszerű rajzolás

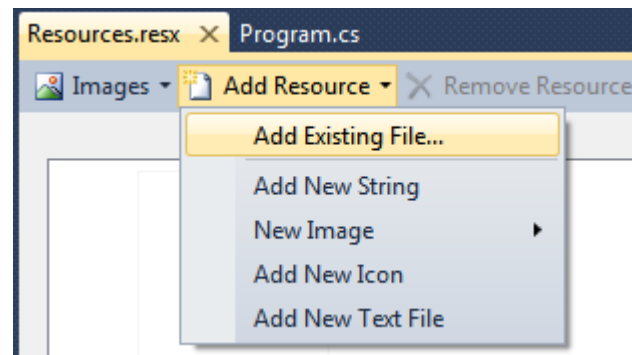
- A Display\_T35 típusú objektum segítségével elérhető a SimpleGraphics példány az alapvető rajzolási műveletekhez:

```
void ProgramStarted()
{
    Debug.Print("Program Started");
    DrawSimpleGraphics();
}

private void DrawSimpleGraphics()
{
    display.SimpleGraphics.BackgroundColor = GT.Color.White;
    display.SimpleGraphics.DisplayEllipse(GT.Color.Cyan, 100, 100, 50, 50);
    display.SimpleGraphics.DisplayRectangle(GT.Color.Red,1,GT.Color.Green, 200, 100, 80, 50);
    display.SimpleGraphics.DisplayText("ABC", Resources.GetFont(Resources.FontResources.NinaB),
    GT.Color.FromRGB(100, 77, 98), 40, 150);
}
```

# SimpleGraphics - Kép rajzolása

- Képek rajzolását úgy végezhetjük el tároló modul nélkül, hogy az erőforrások közé helyezzük azokat:
  1. A Solution Explorer-ben kattintsunk duplán a Resources.resx-re a projektünkön belül.
  2. A megjelenő ablakban különféle erőforrásokat csatolhatunk a programunkhoz.
  3. Kattintsunk az „Add Resource/Add Existing File...” menüpontra.
  4. A felugró ablakban válasszunk ki egy képet és adjunk neki tetszőleges nevet.



# SimpleGraphics - Kép rajzolása

- Fordítás követően a Resources osztály segítségével elérhetővé válik a beágyazott tartalom a megadott név alapján
  - A fordításra azért van szükség, mert a Resources osztályt a fordító generálja a benne lévő erőforrás azonosítókkal (id).
- A kép kijelzőn történő megjelenítését a DisplayImage metódussal lehet elvégezni

```
Bitmap bmp = Resources.GetBitmap(Resources.BitmapResources.head);  
display.SimpleGraphics.DisplayImage(bmp, 0, 0);
```

# Rajzolás bitmap-re

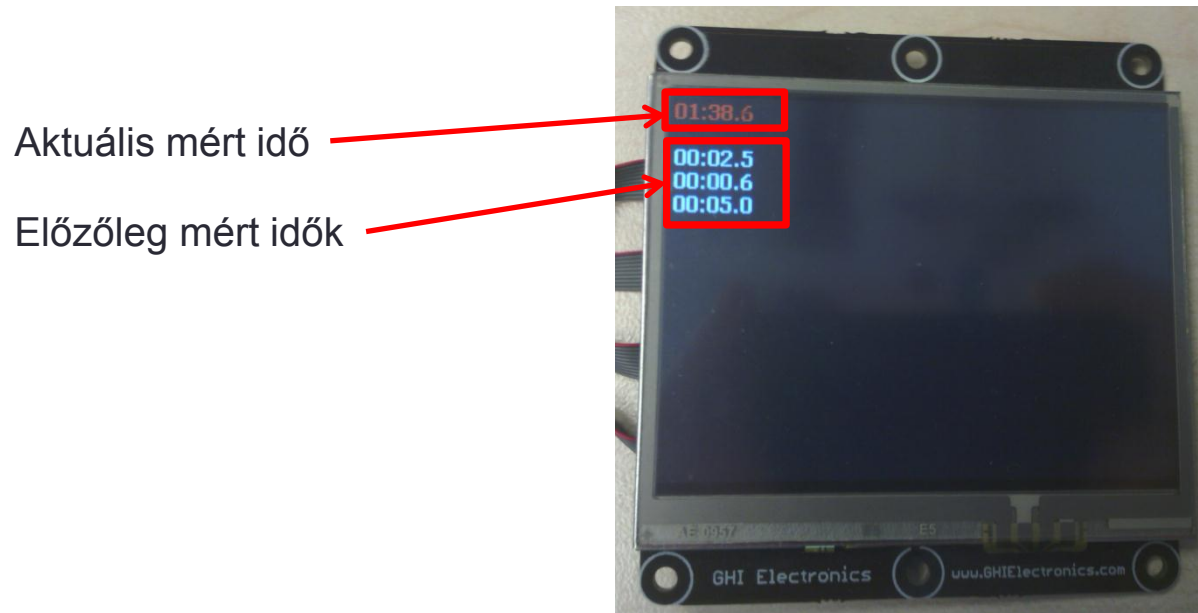
- Rajzolni bitmap objektumra is lehet, majd az megjeleníthető a képernyőn
- Ez a módszer a Doublebuffering alapja

```
private void DrawToBitmap()
{
    int width = (int) display.Width;
    int height = (int) display.Height;
    Bitmap bmp = new Bitmap(width, height);
    bmp.DrawLine(GT.Color.Red, 3, 0, 0, width, height);
    bmp.DrawLine(GT.Color.Red, 3, width, 0, 0, height);
    display.SimpleGraphics.DisplayImage(bmp, 0, 0);
}
```



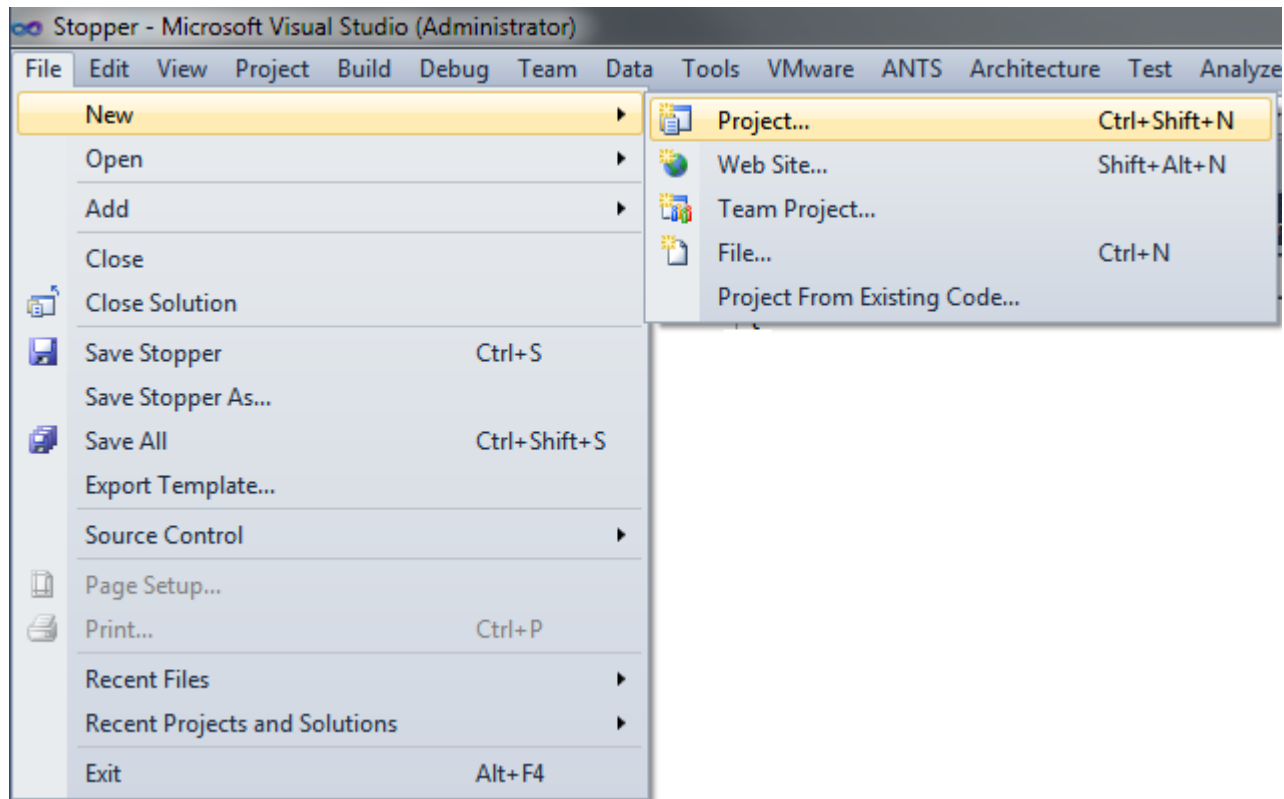
# Feladat - Stopper alkalmazás

- Készítsünk egy egyszerű, grafikus kijelzőt használó stopper alkalmazást:
  - Egy nyomógomb segítségével legyen indítható a stopper
  - Ha az időmérés elindult, a gomb másodszori megnyomásával lehessen leállítani, és az eltelt idő legyen leolvasható
  - A gomb harmadszori megnyomásával a mért idő kerüljön tárolásra, majd a stopper induljon újra
  - A kijelzőn legyen látható az elmúlt három időmérés eredménye



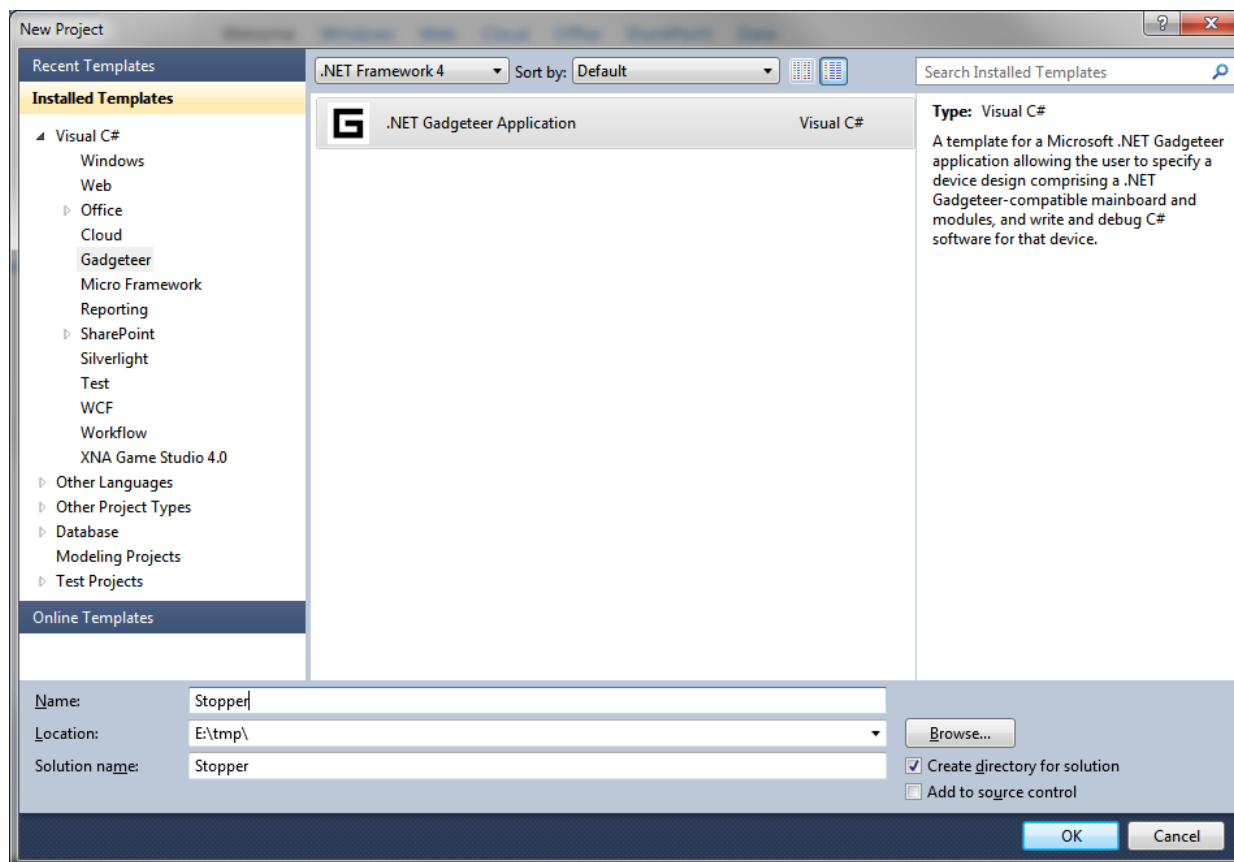
# A projekt létrehozása

- File/New/Project...

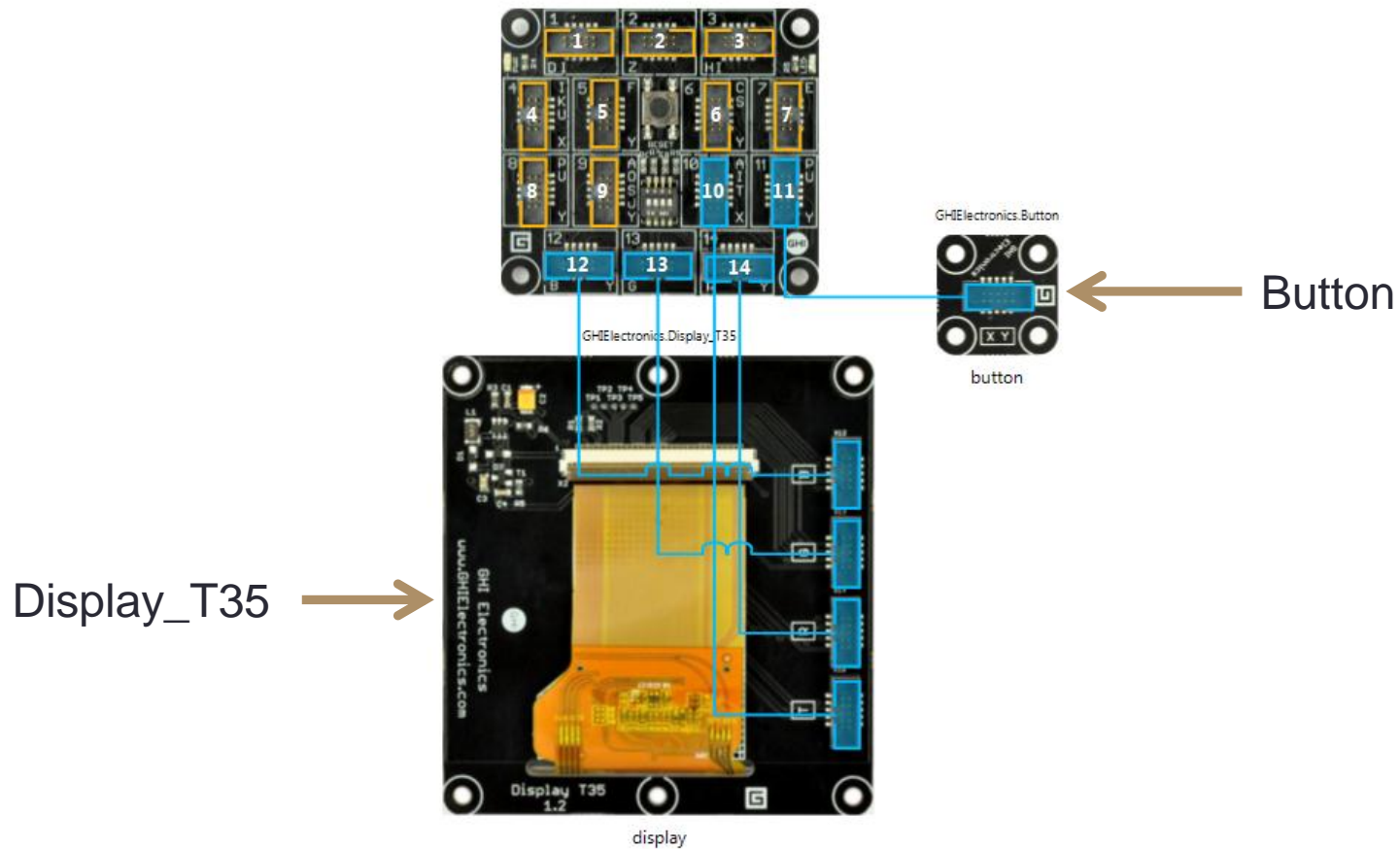


# A projekt létrehozása

- A felugró ablakban válasszuk a Gadgeteer/.NET Gadgeteer Application-t

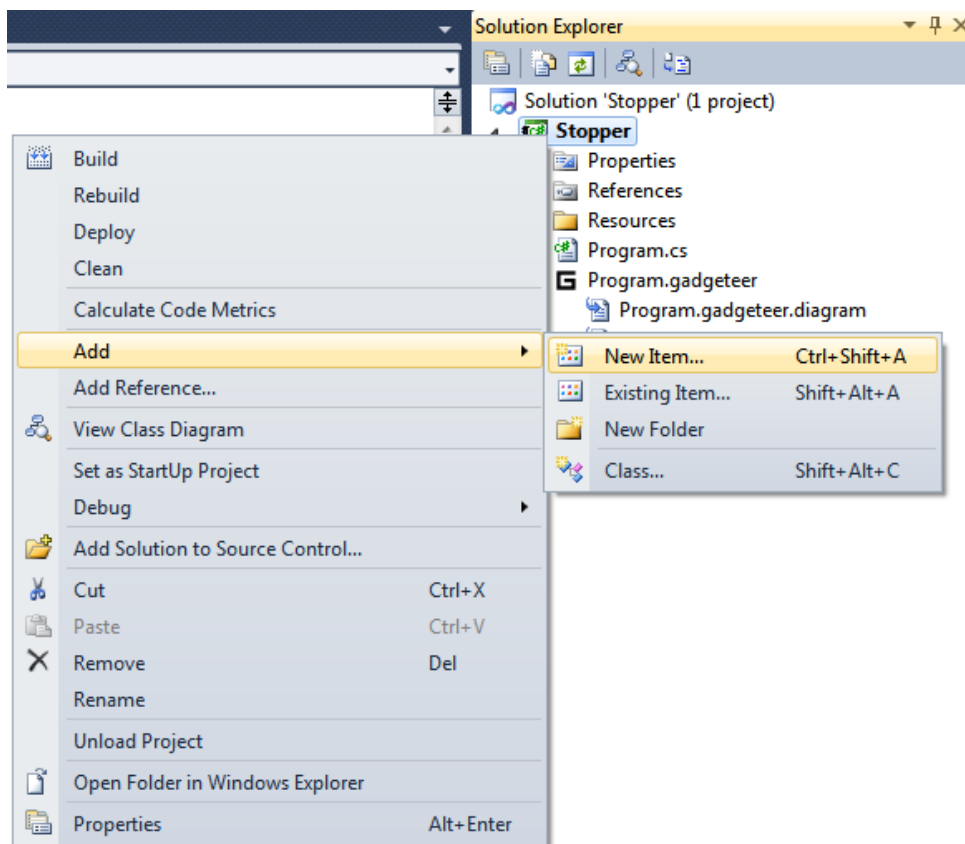


# Stopper készülék elkészítése



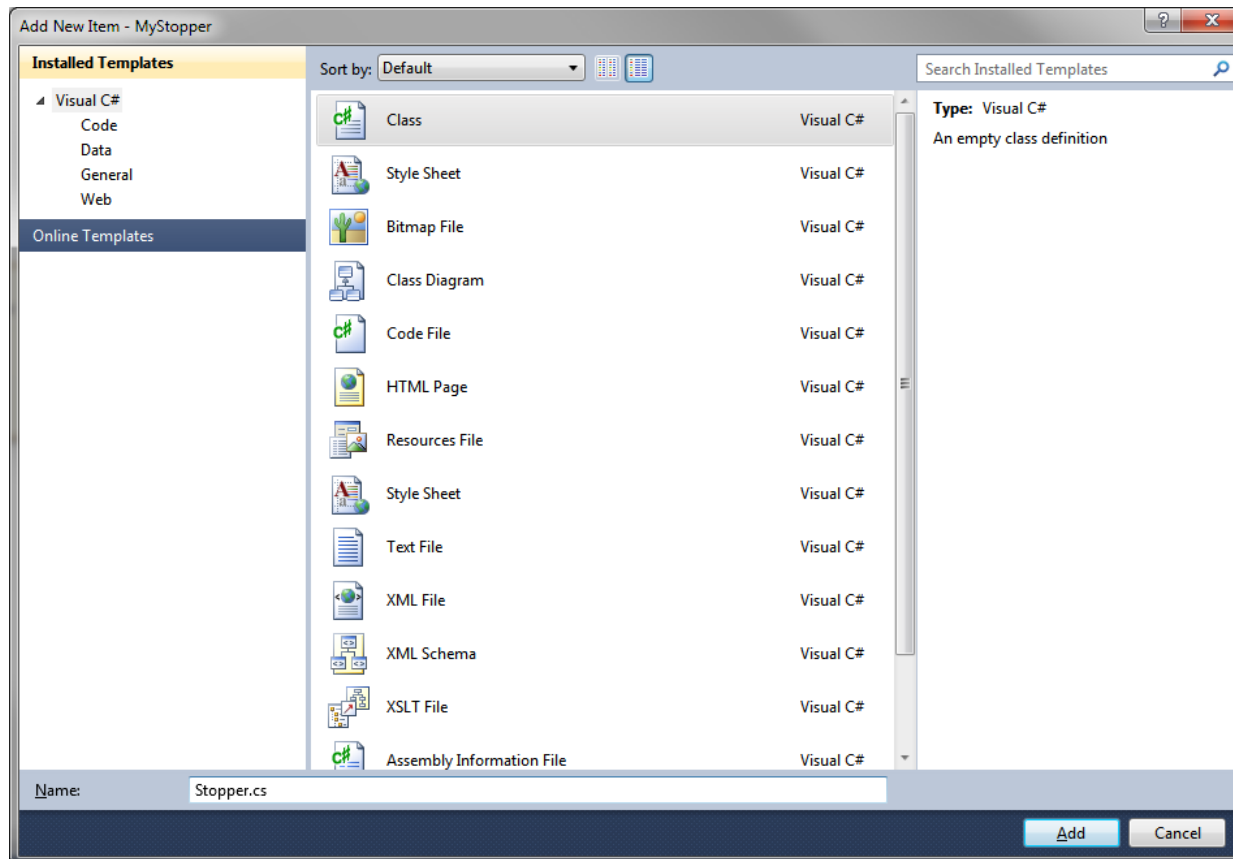
# Új kódfájl hozzáadása a projekthez

- Jobb klikk a projekten/Add/New Item..



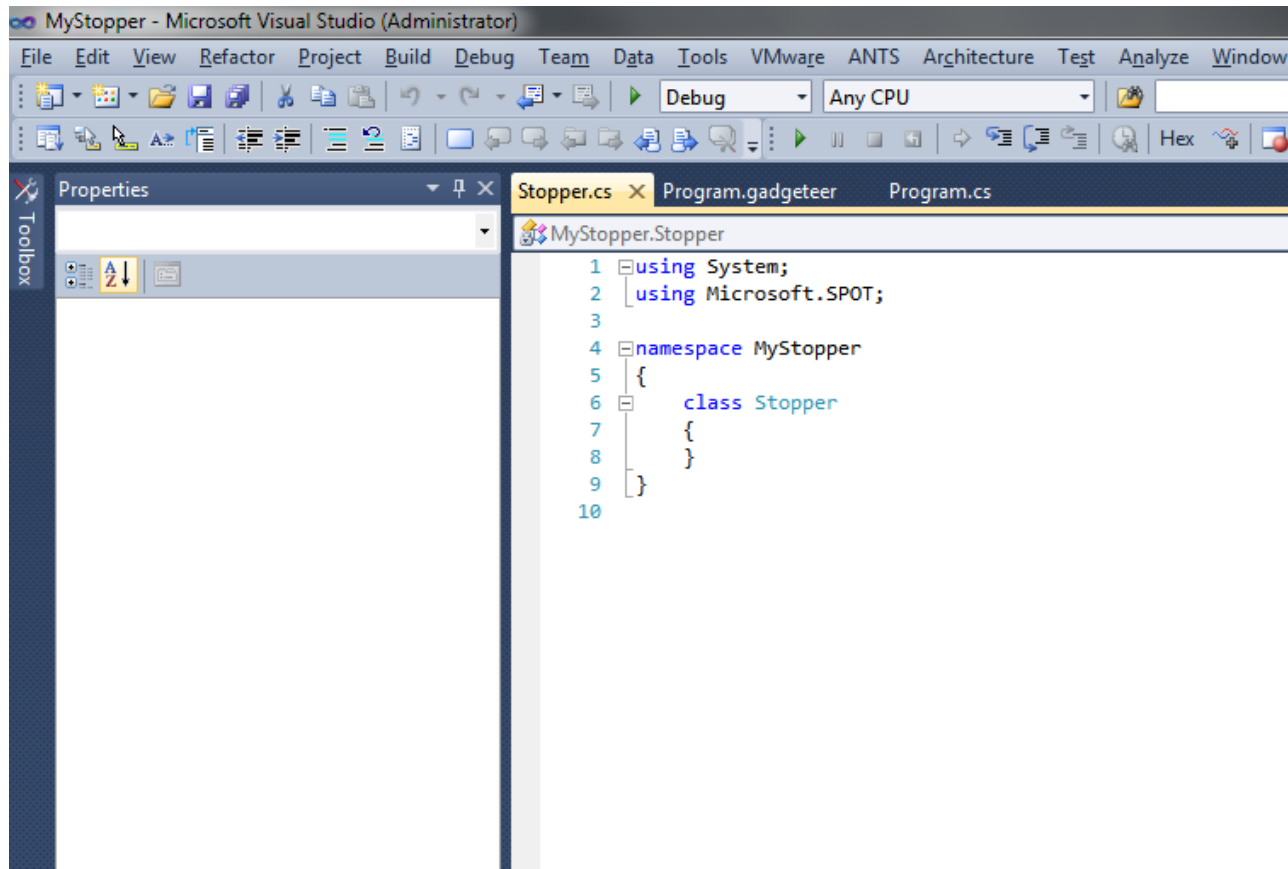
# Új kódfájl hozzáadása a projekthez

- A felugró ablakban válasszuk a „Class”-t:



# Új kódfájl hozzáadása a projekthez

- A létrejött kódfájl:



```
MyStopper - Microsoft Visual Studio (Administrator)
File Edit View Refactor Project Build Debug Team Data Tools VMware ANTS Architecture Test Analyze Window
Debug Any CPU
Properties
Toolbox
Stopper.cs Program.gadgeteer Program.cs
MyStopper.Stopper
1 using System;
2 using Microsoft.SPOT;
3
4 namespace MyStopper
5 {
6     class Stopper
7     {
8     }
9 }
10
```