



Les matériels : la carte Fez BrainPad v1

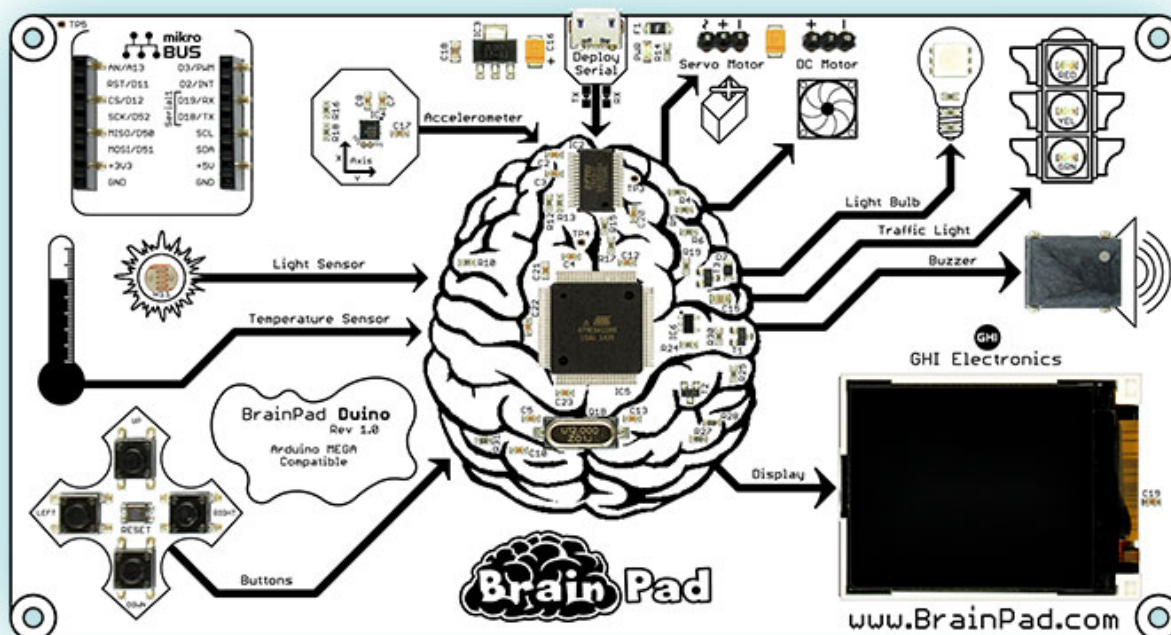
[Mise à jour le 10/1/2019]



Cette carte, programmable en **Scratch**, en **Visual Basic** et en **C#** permet d'aborder la programmation sans se soucier de la configuration de ses entrées/sorties.

Modèle initial

Firmware : .NetMF 4.3 + SDK GHI Electronics (**IDE** : Visual Studio Community 2015)



Principales caractéristiques

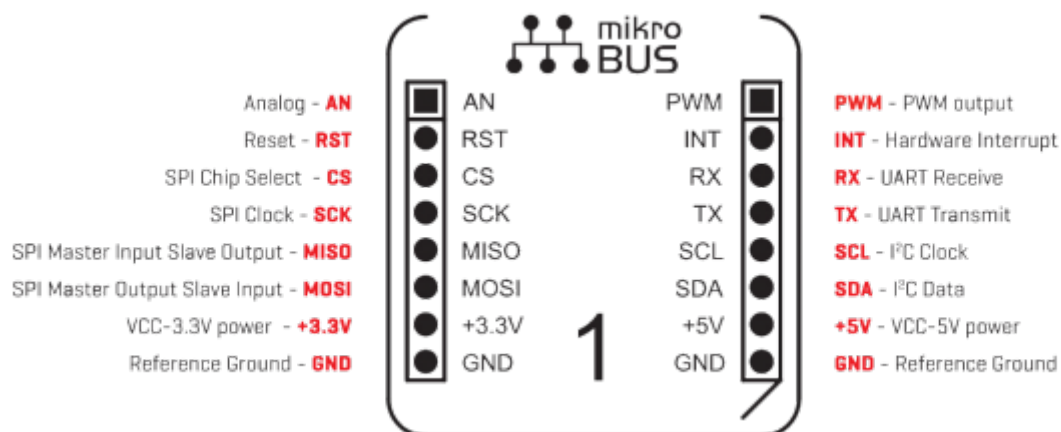
(Extrait du site GHI Electronics)

- **Microcontrôleur** : 84 MHz 32-bit ARM Cortex-M4.
- **SoC** : [G30](#)
- **RAM** : 68 KB.
- **Flash** : 128 KB.
- **Extension** : [mikroBus](#)
- **Sur la carte** : Afficheur graphique (128×160), clavier 4 touches, led RVB, capteurs

(température, luminosité, accéléromètre), buzzer, commande MCC et servomoteur.

G30	G80	G120	G400
G30 System  G30 SoC	G80 System  G80 SoC	G120 System  G120 SoM G120E SoM	G400 System  G400-D SoM G400-S SoM
G30 Duino-Style  FEZ Lemur	G80 Duino-Style  FEZ Panda III	G120 Duino-Style  FEZ Cobra III	G400 Duino-Style <i>Not Available</i>
G30 Through-Hole  G30 TH Module	G80 Through-Hole  G80 TH Module	G120 Through-Hole  G120 TH Module	G400 Through-Hole  G400 TH Module
G30 Dev Board  G30 Dev Board	G80 Dev Board  G80 Dev Board	G120 Dev Board  G120E Dev Board	G400 Dev Board  G400 Dev Board

- **Connecteur d'extension : mikroBUS**



Schéma

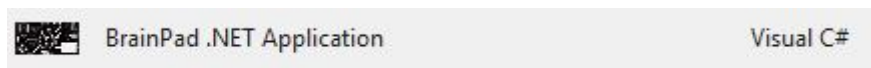
[\[Télécharger\]](#)

Démonstration sur Youtube

[BrainPad : Pour aborder la programmation au lycée Pierre Emile Martin - Bourges](#)

Apperçu du logiciel

Les exemples disponibles dans ce wiki ont été écrits avec l'IDE **Visual Studio Community 2015** enrichi du **microframework .NET v4.3** et du **SDK GHI electronics**. [lien] Lorsque les outils logiciels nécessaires à la carte BrainPad sont installés dans l'environnement de développement intégré (IDE) Visual Studio Community , celui-ci propose un template de projet .



Le code généré par l'IDE Visual Studio lors de la création d'un projet BrainPad est présenté ci-dessous. Son organisation est semblable à celle proposée par l'IDE Arduino. On pourra retenir cette configuration pour débiter et revenir à un code C# plus "classique" par la suite.

Exemple

```
class Program
{
    public void BrainPadSetup()
    {
        //Put your setup code here. It runs once when the BrainPad starts
        up.

        //This code displays "Hello, world!" on the display.
        BrainPad.Display.DrawText(0, 0, "Hello, world!",
        BrainPad.Color.White);
    }

    public void BrainPadLoop()
    {
        //Put your program code here. It runs repeatedly after the BrainPad
        starts up.

        //This code turns the light bulb on and then off.
        BrainPad.LightBulb.TurnOn();
        BrainPad.Wait.Seconds(0.5);
        BrainPad.LightBulb.TurnOff();
    }
}
```

```
BrainPad.Wait.Seconds(0.5);  
}  
}
```

Guide d'installation

Disponible sur le site GHI Electronics [lien](#)

Pour débiter sa programmation

Voir le chapitre [Premiers programmes en C# avec une BrainPad V1 étape par étape](#)

Les classes de la bibliothèque Brainpad

Accessibles à partir de ce [lien](#)

From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

http://webge.fr/dokuwiki/doku.php?id=archives:netmf43:2c_carte_brainpadv1

Last update: **2021/08/11 09:19**

