



TINYCLR OS

Guide de mise en œuvre

RÉSUMÉ

Cet OS est développé par la société **GHI Electronics** sur la base de la **version 4.4** du microframework **.Net**.

Il cible les cartes **FEZ** de cette société (**Panda III, BrainPad, Gadgeteer, Cerberus** etc.), les cartes **MikroElektronika (Quail, Clicker)**, les cartes **Netduino 3** de **WILDERNESS LABS**, mais aussi les cartes à processeur **STM32F4 Discovery board** de **ST Microelectronics**.

Philippe MARIANO

Table des matières

1	Introduction	0
2	Les logiciels à installer sur le PC	1
3	Les logiciels à installer sur la carte cible	2
3.1.1	Le firmware.....	2
3.1.2	Le bootloader.....	4
4	Exemples de code	6
4.1	La carte PANDA III (SoC G80)	7
4.1.1	GPIO – Blink : une première application sous TinyCLR-OS	7
4.1.2	GPIO - le BP Ldr0 de la carte commande la Led1.....	8
4.1.3	GPIO - Clavier 4T Digilent connecté à une carte Tinkerkit.....	9
4.1.4	PWM - Faire varier la luminosité de la Led1 de la carte	10
4.1.5	ANALOG - PWM - Changer la fréquence d'un signal PWM (Buzzer) avec un potentiomètre.....	11
4.1.6	Commander un moteur pas à pas avec un module EasyStepper	12
4.1.7	SPI - Commande d'un afficheur graphique ADAFRUIT 160 x 140 à ST7735R	14
4.2	Les cartes BrainPad	16
4.2.1	Procédures d'installation (Sept 2018).....	16
4.2.2	Mise en œuvre d'un programme sur la carte BrainPad V2 sous MakeCode	16
4.2.3	La carte BRAINPAD v1 (SoC G30)	17
4.2.4	La carte BRAINPAD v2 (STM32F401RE)	18
4.3	La carte NETDUINO 3	27
4.3.1	GPIO - Blink	27
Annexe 1 : L'outil TinyCLR Config		28
Annexe 2 : Résolution du problème d'installation du driver de la carte sous Windows 7		29

Site de documentation GHI Electronics

<https://goo.gl/51v1vX>

TinyCLR OS Version 1.0.0 (dernière version mise en ligne le 29/12/2018 sur le site de [GHI Electronics](https://www.gih.com))

Dernière révision de ce document le 29/12/2018

1 Introduction

L'OS TinyCLR est le futur des cartes à microcontrôleur programmables en C#. Il est développé par la société **GHI ELECTRONICS** . C'est un "système d'exploitation" pour l'embarqué, construit à partir de .NETMF 4.4 et déployé avec des paquets NuGet. Les anciens projets .NETMF v4.3 sont toujours utilisables mais nécessitent quelques modifications dans la gestion des entrées, sorties.

Avantages par rapport au .NetMF de Microsoft (plus maintenu depuis 2015)

- Plus de SDK à installer, mais des paquets nuget
- Bibliothèques spécifiques aux matériels utilisés
- Syntaxe de programmation similaire à W10 IoT pour Raspberry Pi

2 Les logiciels à installer sur le PC

- **Visual Studio**

1. Installer l'environnement de développement intégré [VISUAL STUDIO COMMUNITY 2017](#). TinyCLR OS ne fonctionne pas avec les précédentes antérieures.



La procédure à suivre pour installer la dernière version de l'**extension GHI pour VS2017 (§2)**, les **bibliothèques de code (§3)** et le **firmware** de la carte cible (§4) est détaillée ci-dessous.

Cette procédure est tirée du site de GHI Electronics.

<http://bit.ly/2KcQyU7>

- **Les logiciels de GHI Electronics**



2. **Téléchargez et installez** la dernière version de l'extension **Visual Studio Project System** pour l'IDE **Visual Studio 2017** à partir du lien [DOWNLOADS](#) disponible sur la page "[TinyCLR/Introduction](#)". Ce lien renvoie vers la page [Downloads](#) où sont répertoriées les différentes versions de cette extension.

Exemple

Visual Studio Project System

The extension is what gets loaded on Visual Studio to allow it to communicate with a TinyCLR OS device. It also includes project templates.

File	Date	Status	MD5
v0.6.0	2017-08-31	Alpha	68E94D0720CC0AC1A3E98DEF7704FAAA

3. **Téléchargez** la dernière version des **bibliothèques**.

Exemple

Libraries

Libraries are hosted through a local NuGet feed for now as we are a lot of changes still. These libraries will be hosted on <http://www.nuget.org/> in the future.

File	Date	Status	MD5
v0.6.0	2017-08-31	Alpha	CA9275032B3A2EA403738497C2C0C280

4. **Copiez** ces bibliothèques dans un répertoire (par exemple Tinyclr C:\Users\phili\OneDrive\Documents\Visual Studio 2017\Tinyclr\TinyCLR.0.x.0) pour en faire un **NuGet local**.

Exemple

- GHIElectronics.TinyCLR.BrainPad.0.6.0.nupkg
- GHIElectronics.TinyCLR.Core.0.6.0.nupkg
- GHIElectronics.TinyCLR.Devices.0.6.0.nupkg
- GHIElectronics.TinyCLR.Drawing.0.6.0.nupkg
- GHIElectronics.TinyCLR.Pins.0.6.0.nupkg
- GHIElectronics.TinyCLR.Storage.0.6.0.nupkg

Les bibliothèques seront sur **Nuget.org** dans le futur !

3 Les logiciels à installer sur la carte cible

3.1.1 Le firmware

Il doit être réinstallé à chaque changement de version de l'extension "**Visual Studio Project System**" pour **VS2017**.

IMPORTANT depuis la version v0.6.0, l'outil **TinyCLR Config**, situé dans le **menu GHI Electronics**, permet de mettre à jour le firmware des cartes chargées avec le **GHI Bootloader v2 à condition qu'un OS soit présent (voir l'annexe 1)**. Pour celle disposant d'un Bootloader V1, ou d'une version antérieure de TinyCLR OS, suivre l'exemple de procédure donné ci-dessous pour les cartes **Fez PANDA 3**.

Les explications données ci-dessous sont issues du [Teck Talk #041](#) - Vidéo "**Updating BrainPad's Firmware**". La démarche est applicable à toutes les cartes.

<http://bit.ly/2IYjIDj>

Préalable : **Téléchargez Tera Term** (pas d'installation)



Procédure

- a. **Téléchargez** le firmware de la carte ciblée à partir de la page [Downloads](#).
- Pour la carte **Panda 3**, le processeur est un G80 comme dans l'exemple ci-dessous.

G80

File	Date	Status	MDS
v0.7.0	2018-01-04	Alpha	38738974801871CE81894CA0C9759C0A

- Pour la carte **BrainPad v2**, sélectionner FEZCLR.

FEZCLR

This is the reference firmware for TinyCLR OS.

File	Date	Status	MDS
v0.12.0	2018-07-05	Alpha	26863FE03DB65A3EC3D238E83A6220DE

- b. **Connectez** la carte au PC et la placer en mode boot loader.

- Pour la carte Panda 3**

Action simultanée sur les boutons-poussoirs LDR0 et LDR1, suivi d'une action sur le bouton Reset puis relâchez LDR0 et LDR1. La carte doit être accessible par un port COMx (Ex : COM5) dans "Périphériques et imprimantes".



- Pour la carte BrainPad BP2**

Action simultanée sur les boutons-poussoirs L et Reset, relâcher Reset puis L.

Particularités de cette carte : Elle peut être programmée avec [BrainPad Makecode](#) ou Visual Studio Community. Pour cela, elle est chargée avec le bootloader [v1.0.2](#). L'action sur les boutons L et Reset ouvre une fenêtre dans laquelle il suffit de faire glisser le fichier produit par Makecode ou le [firmware](#) spécifique à cette carte (programmation sous MSVS)

TinyCLR OS BrainPad Firmware

This tiny operating system can be loaded onto the BrainPad to enable it to work with Visual Studio.

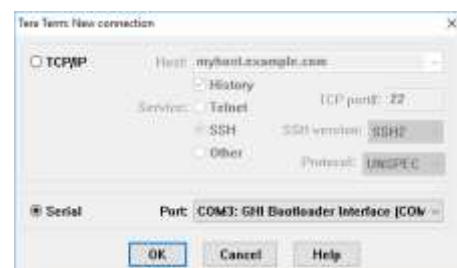
File	Date	MDS
v1.0.0-preview2	2018-11-30	E2E981786382DCB13DEB3E43CCBF864E

- Autres cartes** : voir la vidéo [Teck Talk #041](#).

- c. **Lancez Tera Term** (ttermpro.exe).

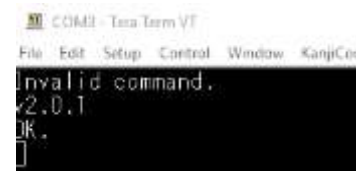
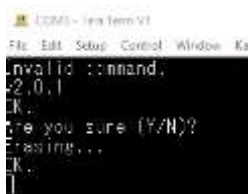
- d. **Sélectionnez Serial** et le port sur lequel apparaît la carte. Puis OK.

- e. **Entrez v** puis **[Entrée]** dans la console qui vient de s'ouvrir **pour visualiser** la version du bootloader. Elle doit être égale à v2.0.1 ou supérieure sinon voir (**\$ Le bootloader**) ou la vidéo [Teck Talk #041](#) pour le recharger.



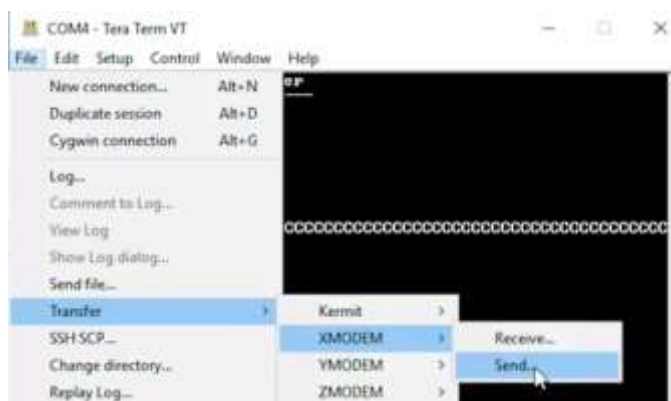
- f. **Effacez** la mémoire
Entrer **e** puis **[Entrée]** puis **Yes et [Entrée]** à la question Are you sure (Y/N).

La console présente alors un message indiquant que l'effacement a été correctement réalisé comme dans la copie d'écran ci-dessous.



- g. **Mise à jour** du firmware.
Pour établir la connexion avec la carte, entrez **x** (ou **u**, voir la vidéo !) puis **[Entrée]** puis **Yes et [Entrée]** à la question Are you sure (Y/N). Une série de CCCCCCCCCC s'affiche sur l'écran indiquant que la connexion est établie.

Il faut maintenant sélectionner le firmware à télécharger. Ceci est réalisé en sélectionnant le menu "Send..." comme ci-dessous.



IMPORTANT : Dans la fenêtre qui s'ouvre, sélectionner **1K**.

Sélectionnez le firmware à télécharger à son emplacement sur le disque. Par exemple *G80 Firmware.x.x.x.ghi* pour la carte Panda 3.

Cliquez sur **Ouvrir**. Si tout se passe bien !



Pour les autres cartes, se reporter à la documentation en ligne sur docs.ghielectronics.com

Exemple : Installation du firmware sur une carte Netduino 3 (STM32F)

<http://bit.ly/2u6mwfk>

3.1.2 Le bootloader

Le **bootloader** est réinstallé exceptionnellement (en cas de problème ou si une **mise à jour est nécessaire**). Un exemple de procédure est donné ci-dessous pour la carte **BrainPad BP2**.

Les explications ci-dessous sont issues du [Teck Talk #041](#) -Vidéo "Updating BrainPad's Firmware". La démarche est applicable à toutes les cartes.

<http://bit.ly/2lYlDj>

- a) **Téléchargez** le binaire à transférer dans la carte BrainPad BP2 à partir du site de GHI Electronics.

<http://bit.ly/2v99tvp>

Bootloader

This is the TinyCLR OS BrainPad bootloader. See the DFU Files page for instructions on loading DFU files.

File	Date	MD5
v1.0.2	2018-04-05	4285FFAD09CF4C4C1B3C676FBC53807A

- b) Pour télécharger le bootloader, il est nécessaire de passer la carte en mode DFU, pour cela :
- Appuyer sur le bouton **BOOT0** de la carte BrainPad et le maintenir
 - Appuyer sur le bouton **Reset** et le relâcher.
 - Relâcher le bouton **BOOT0**.

L'icône ci-contre doit apparaître dans "Périphériques et imprimantes".



- c) **Téléchargez** l'outil **DfuSe** v3.05 ou supérieur sur le site de [ST Microélectronics](#) et l'installer

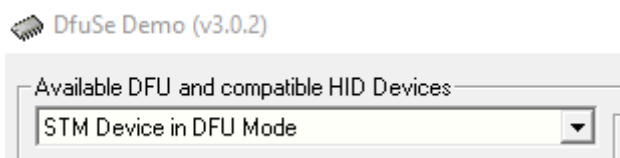
<http://bit.ly/2u9cjzi>

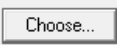


- d) **Lancez**

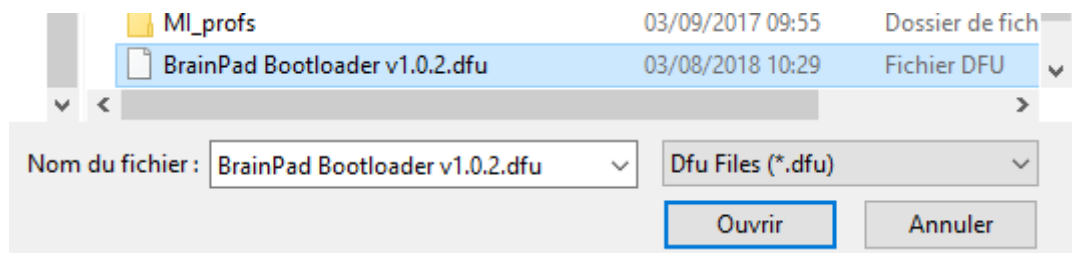


- e) **Réglez** "Available DFU and compatible HID Devices" comme ci-dessous.



- f) Cliquez sur  pour sélectionner le fichier préalablement téléchargé sur le site de GHI Electronics.

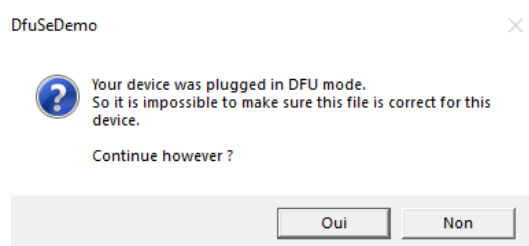
Exemple pour une carte BrainPad BP2



Si tout se passe correctement !



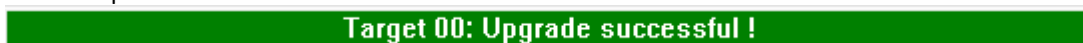
- g) Cliquez sur  pour télécharger le bootloader.



Cliquez sur Oui

Une barre de téléchargement doit indiquer sa progression. Faire ensuite un reset de la carte.

Si tout se passe correctement !



4 Exemples de code

- Exemples sur le site [GHI Electronics](http://www.ghi.com)







<http://bit.ly/2u6Nzrp>

- Créer un projet avec Visual Studio Community 2017

Nouveau projet

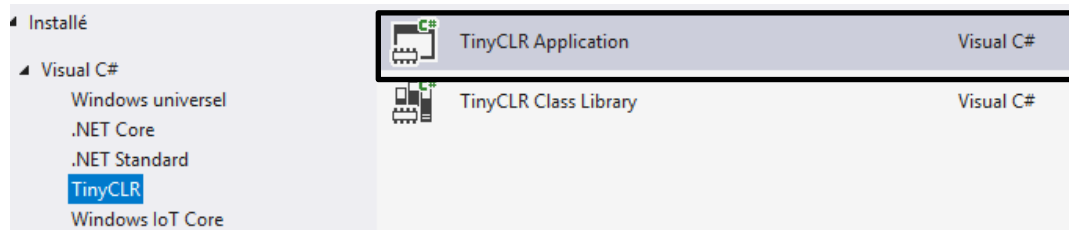
Rechercher dans les modèles de projet

Modèles de projet récents :



	TinyCLR Application	C#
	TinyCLR Application	C#
	Application console (.NET Core)	C#
	Application console (.NET Framework)	C#
	TinyCLR Application	C#
	TinyCLR Class Library	C#

[Créer un projet...](#)

Choisir le template ci-dessous



The screenshot shows the Visual Studio 'New Project' dialog. On the left, a tree view shows 'Installé' expanded to 'Visual C#', with sub-items: 'Windows universel', '.NET Core', '.NET Standard', 'TinyCLR' (highlighted in blue), and 'Windows IoT Core'. On the right, a list of templates is shown. The 'TinyCLR Application' template is highlighted with a black box, showing its icon, name, and 'Visual C#' language. Below it, the 'TinyCLR Class Library' template is also visible.

Installé		
Visual C#		
Windows universel		
.NET Core		
.NET Standard		
TinyCLR		TinyCLR Application Visual C#
Windows IoT Core		TinyCLR Class Library Visual C#

4.1 La carte PANDA III (Soc G80)

4.1.1 GPIO – Blink : une première application sous TinyCLR-OS

Entrées, sorties à usage général

Un objet de la classe **GPIOPin** correspond à une broche GPIO. Pour accéder à une telle broche, il faut d'abord l'ouvrir. On le fait en utilisant la méthode **OpenPin** d'un objet de la classe **GpioController**. La broche peut être paramétrée en entrée, en sortie, avec résistance de pull-up, etc. en appliquant la méthode **SetDriveMode**.

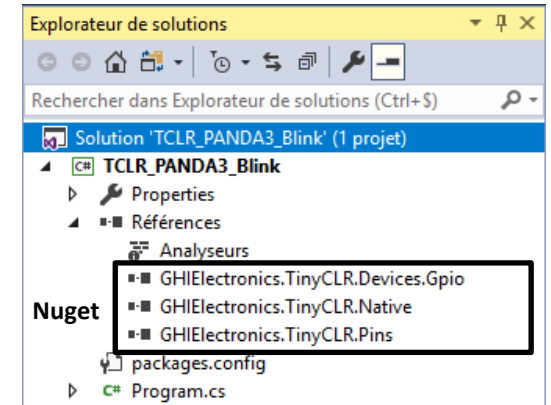
Code C#

```
using System.Threading;
using GHIElectronics.TinyCLR.Pins;
using GHIElectronics.TinyCLR.Devices.Gpio;

namespace TCLR_PANDA3_Blink
{
    public class Program
    {
        public static void Main()
        {
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/gpio.html
            // Get the default GPIO controller on the system
            var gpio = GpioController.Default; // (1)

            // Open GPIO Led1
            GpioPin led1 = gpio.OpenPin(FEZPandaIII.GpioPin.Led1); // (2)
            // Set the IO direction as output
            led1.SetDriveMode(GpioPinDriveMode.Output); // (3)

            while (true)
            {
                led1.Write(GpioPinValue.High);
                Thread.Sleep(100);
                led1.Write(GpioPinValue.Low);
                Thread.Sleep(500);
            }
        }
    }
}
```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

Résumé

1. Utiliser **GpioController.Default** () pour obtenir le contrôleur GPIO. Si le circuit ne dispose pas de contrôleur GPIO, cette méthode retourne null.
2. Ouvrir la broche en appelant **gpio.OpenPin** () avec la valeur **FEZPandaIII.GpioPin.Led1**.
3. Paramétrer la broche (sens, résistance de rappel, etc.).

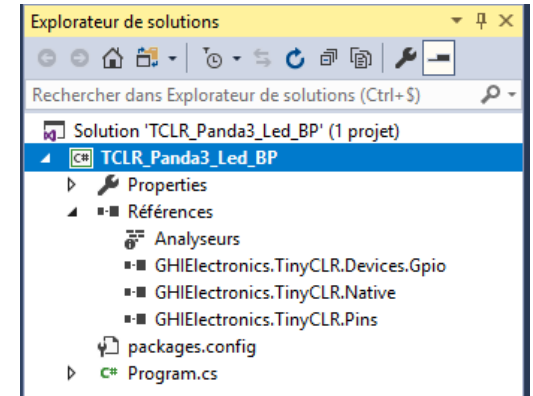
4.1.2 GPIO - le BP "Ldr0" de la carte commande la led "Led1"

Code C#

```
using System.Threading;
using GHIElectronics.TinyCLR.Devices.Gpio;
using GHIElectronics.TinyCLR.Pins;

namespace TCLR_Panda3_Led_BP
{
    public class Program
    {
        public static void Main()
        {
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/gpio.html
            // Get the default GPIO controller on the system
            var gpio = GpioController.GetDefault();
            // Open GPIO Led1
            GpioPin led = gpio.OpenPin(FEZPandaIII.GpioPin.Led1);
            // Set the IO direction as output
            led.SetDriveMode(GpioPinDriveMode.Output);
            // Open GPIO BP
            GpioPin bp = gpio.OpenPin(FEZPandaIII.GpioPin.Ldr0);
            // Set the IO direction as input
            bp.SetDriveMode(GpioPinDriveMode.InputPullUp);

            while (true)
            {
                if (bp.Read() == GpioPinValue.High)
                {
                    led.Write(GpioPinValue.Low);
                    //Debug.WriteLine("BP activé");
                }
                else
                {
                    led.Write(GpioPinValue.High);
                    //Debug.WriteLine("BP relâché");
                }
                Thread.Sleep(10);
            }
        }
    }
}
```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

4.1.3 GPIO - Clavier 4T Digilent connecté à une carte Tinkerkit

Code C#

```

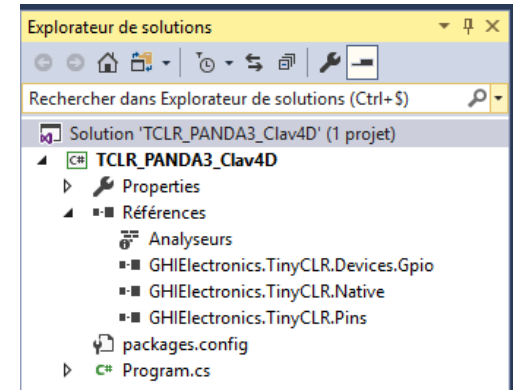
using System.Threading;
using GHIElectronics.TinyCLR.Devices.Gpio;
using GHIElectronics.TinyCLR.Pins;

namespace TCLR_PANDA3_Clav4D
{
    public class Program
    {
        public static void Main()
        {
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/gpio.html
            // Get the default GPIO controller on the system
            var gpio = GpioController.Default;
            // Open GPIO Led1
            GpioPin led1 = gpio.OpenPin(FEZPandaIII.GpioPin.Led1);
            // Set the IO direction as output
            led1.SetDriveMode(GpioPinDriveMode.Output);
            // Open GPIO BTN0
            GpioPin BTN0 = gpio.OpenPin(FEZPandaIII.GpioPin.A0);
            // Set the IO direction as input
            BTN0.SetDriveMode(GpioPinDriveMode.Input);
            // Open GPIO BTN1
            GpioPin BTN1 = gpio.OpenPin(FEZPandaIII.GpioPin.A1);
            // Set the IO direction as input
            BTN1.SetDriveMode(GpioPinDriveMode.Input);
            // Open GPIO BP2
            GpioPin BTN2 = gpio.OpenPin(FEZPandaIII.GpioPin.A2);
            // Set the IO direction as input
            BTN2.SetDriveMode(GpioPinDriveMode.Input);
            // Open GPIO BTN3
            GpioPin BTN3 = gpio.OpenPin(FEZPandaIII.GpioPin.A3);
            // Set the IO direction as input
            BTN3.SetDriveMode(GpioPinDriveMode.Input);

            while (true)
            {
                if ((BTN0.Read() == GpioPinValue.High) || (BTN1.Read() == GpioPinValue.High ) || (BTN2.Read() == GpioPinValue.High) || (BTN3.Read() == GpioPinValue.High))
                {
                    led1.Write(GpioPinValue.High);
                }
                else
                {
                    led1.Write(GpioPinValue.Low);
                }

                Thread.Sleep(100);
            }
        }
    }
}

```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

4.1.4 PWM - Faire varier la luminosité de la Led1 de la carte

Code C#

```
using System.Diagnostics;
using System.Threading;
using GHIElectronics.TinyCLR.Devices.Pwm;
using GHIElectronics.TinyCLR.Pins;

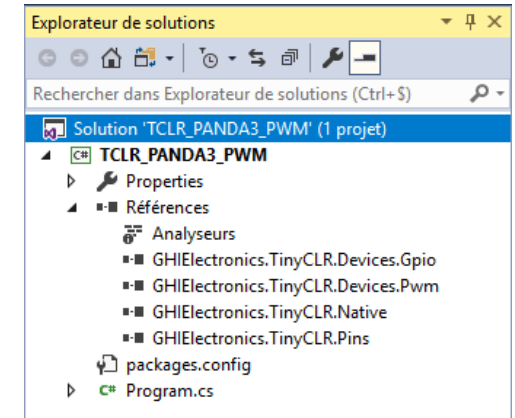
namespace TCLR_PANDA3_PWM
{
    public class Program
    {
        public static void Main()
        {
            double step = 0.01;
            double level = 0.5; // Rapport cyclique

            // PWM
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/pwm.html
            // Make sure you only open pins from the same controller the controller was constructed for, in this case Controller
            var controller = PwmController.FromId(FEZPandaIII.PwmPin.Controller1.Id);
            var led1 = controller.OpenPin(FEZPandaIII.PwmPin.Controller1.Led1);

            controller.SetDesiredFrequency(1000);

            led1.Start();

            while (true)
            {
                level = level + step;
                led1.SetActiveDutyCyclePercentage(level);
                if ((level >= 0.9) || (level <= 0.1))
                {
                    step = step * -1; // Invert the step
                }
                Debug.WriteLine(level.ToString("F2"));
                Thread.Sleep(10);
            }
        }
    }
}
```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

Résumé

1. Obtenir un contrôleur PWM
2. Paramétrer le contrôleur (fréquence, rapport cyclique, etc.)
3. Ouvrir la broche

4.1.5 ANALOG - PWM - Changer la fréquence d'un signal PWM (Buzzer) avec un potentiomètre

Code C#

```
using System;
using System.Threading;
using GHIElectronics.TinyCLR.Pins;
using GHIElectronics.TinyCLR.Devices.Adc;
using GHIElectronics.TinyCLR.Devices.Pwm;

namespace TCLR_PANDA3_POT_PWM
{
    public class Program
    {
        public static void Main()
        {
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/pwm.html
            // http://docs.ghielectronics.com/software/tinyclr/tutorials/adc.html
            AdcController ADC = AdcController.Default();
            AdcChannel A0 = ADC.OpenChannel(FEZPandaIII.AdcChannel.A0);

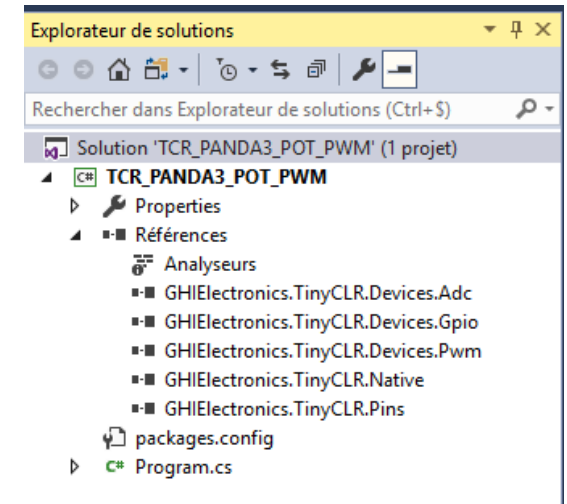
            PwmController controllerPWM = PwmController.FromId(FEZPandaIII.PwmPin.Controller11.Id);
            PwmPin Buzz = controllerPWM.OpenPin(FEZPandaIII.PwmPin.Controller11.D5);

            double lastAnalog = A0.ReadRatio();

            while (true)
            {
                double delta = Math.Abs(lastAnalog - A0.ReadRatio());

                if (delta > 0.01)
                {
                    lastAnalog = A0.ReadRatio();
                    Buzz.Stop();
                    controllerPWM.SetDesiredFrequency((lastAnalog * 3000) + 300);
                    Buzz.Start();
                    Buzz.SetActiveDutyCyclePercentage(0.5);

                    //System.Diagnostics.Debug.WriteLine("A0 " + (int)(A0.ReadRatio() * 100));
                }
                Thread.Sleep(50);
            }
        }
    }
}
```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

4.1.6 Commander un moteur pas à pas avec un module EasyStepper

Code C#

```
using System;

using System.Diagnostics;
using System.Threading;
using GHIElectronics.TinyCLR.Pins;

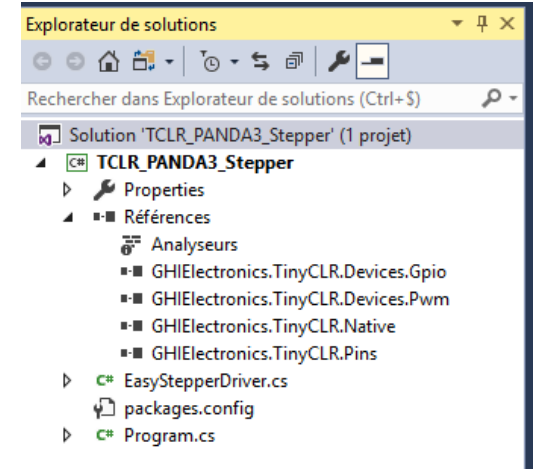
namespace TCLR_PANDA3_STEPPER
{
    class Program
    {
        static void Main()
        {
            // Test for stepper motor ITC-VNC-2 N=200steps - U=12V - C=200g/cm
            var time = 2000; UInt16 delay = 5; UInt32 nbpas = 200;
            var stepper = new EasyStepperDriver(FEZPandaIII.GpioPin.D13, FEZPandaIII.GpioPin.D12,
                FEZPandaIII.GpioPin.D2, FEZPandaIII.GpioPin.D10, FEZPandaIII.GpioPin.D11, FEZPandaIII.GpioPin.D3);

            stepper.WakeUp();

            while (true)
            {
                // Using Step(), StepMode, StepDirection and Steptime
                stepper.EnableOutputs();
                Debug.WriteLine("Sleep= " + stepper.IsDriverSleep + " Enable= " + stepper.IsOutputsEnable);
                Debug.WriteLine("Full Forward"); // 360° for ITC-VNC-1 motor
                stepper.Turn(nbpas, EasyStepperDriver.Direction.Forward, delay, EasyStepperDriver.Mode.Full);
                Debug.WriteLine("Steps= " + stepper.Steps + " Mode= " + stepper.StepMode + " Dir= " + stepper.StepDirection + " time= " + stepper.StepDelay
+ "ms" + "\n");
                stepper.DisableOutputs(); Thread.Sleep(time);

                Debug.WriteLine("Half Backward"); stepper.EnableOutputs(); // 180° for ITC-VNC-1 motor
                stepper.Turn(nbpas, EasyStepperDriver.Direction.Backward, delay, EasyStepperDriver.Mode.Half);
                Debug.WriteLine("Steps= " + stepper.Steps + " Mode= " + stepper.StepMode + " Dir= " + stepper.StepDirection + " time= " + stepper.StepDelay
+ "ms" + "\n");
                stepper.DisableOutputs(); Thread.Sleep(time);

                Debug.WriteLine("Quater Forward"); stepper.EnableOutputs(); // 90° for ITC-VNC-1 motor
                stepper.Turn(nbpas, EasyStepperDriver.Direction.Forward, delay, EasyStepperDriver.Mode.Quarter);
                Debug.WriteLine("Steps= " + stepper.Steps + " Mode= " + stepper.StepMode + " Dir= " + stepper.StepDirection + " time= " + stepper.StepDelay
+ "ms" + "\n");
                stepper.DisableOutputs(); Thread.Sleep(time);
            }
        }
    }
}
```



```

Debug.WriteLine("OneEighth Backward"); stepper.EnableOutputs(); // 45° for ITC-VNC-1 motor
stepper.Turn(nbpas, EasyStepperDriver.Direction.Backward, 1, EasyStepperDriver.Mode.OneEighth);
Debug.WriteLine("Steps= " + stepper.Steps + " Mode= " + stepper.StepMode + " Dir= " + stepper.StepDirection + " time= " + stepper.StepDelay
+ "ms" + "\n");
stepper.DisableOutputs(); Thread.Sleep(3 * time);
    }
}
}
}
}

```



Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

4.1.7 SPI - Commande d'un afficheur graphique ADAFRUIT 160 x 140 à ST7735R (partiel)

```

using GHIElectronics.TinyCLR.Devices.Display;
using GHIElectronics.TinyCLR.Devices.Display.Provider;
using GHIElectronics.TinyCLR.Devices.Gpio;
using GHIElectronics.TinyCLR.Devices.Spi;
using GHIElectronics.TinyCLR.Drawing;
using GHIElectronics.TinyCLR.Pins;
using System;
using System.Drawing;
using System.Threading;

namespace TCLR_PANDA3_AdaDisplayShield
{
    class Program
    {
        static void Main()
        {
            var spi = SpiController.FromName(G80.SpiBus.Spi1);
            var gpio = GpioController.GetDefault();
            var st7735 = new ST7735Controller(spi.GetDevice(ST7735Controller.GetConnectionSettings(SpiChipSelectType.Gpio,
FEZPandaIII.GpioPin.D10)), gpio.OpenPin(FEZPandaIII.GpioPin.D8), gpio.OpenPin(FEZPandaIII.GpioPin.D9));

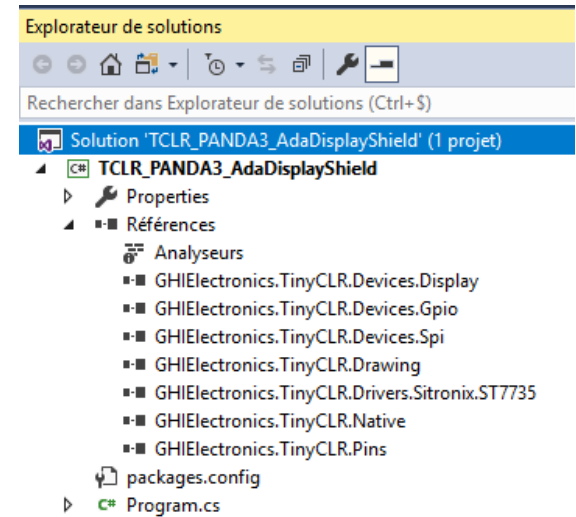
            var disp = DisplayController.FromProvider(st7735);
            disp.SetConfiguration(new SpiDisplayControllerSettings { Width = 160, Height = 128 });

            var bl = gpio.OpenPin(G80.GpioPin.PC7);
            bl.Write(GpioPinValue.High);
            bl.SetDriveMode(GpioPinDriveMode.Output);

            var hdc = GraphicsManager.RegisterDrawTarget(new DrawTarget(disp));
            var screen = Graphics.FromHdc(hdc);

            var rnd = new Random();
            var x = rnd.Next(160);
            var y = rnd.Next(128);
            var vx = rnd.Next(20) - 10;
            var vy = rnd.Next(20) - 10;
            var color = new Pen(Color.White);

```



[Adafruit 1.8" Color TFT Shield w/microSD and Joystick](#)

```

while (true)
{
    x += vx;
    y += vy;


















    if (x >= 160 || x < 0) vx *= -1;
    if (y >= 128 || y < 0) vy *= -1;

    screen.Clear(Color.Black);
    screen.DrawEllipse(color, x, y, 10, 10);
    screen.Flush();

    Thread.Sleep(10);
}
}
}

```

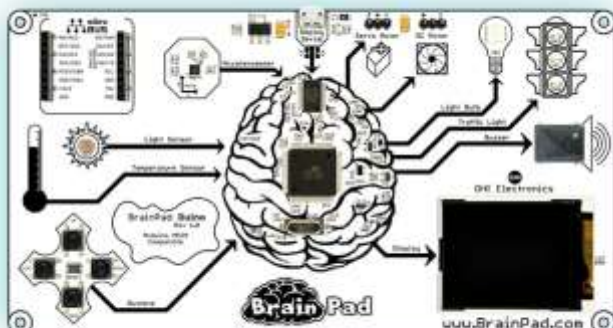


	GHIElectronics.TinyCLR.Core par GHI Electronics, LLC	 v1.0.0 
	The mscorlib library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Devices.Display par GHI Electronics, LLC	 v1.0.0
	The display library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Devices.Gpio par GHI Electronics, LLC	 v1.0.0
	The GPIO library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Devices.Spi par GHI Electronics, LLC	 v1.0.0
	The SPI library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Drawing par GHI Electronics, LLC	 v1.0.0
	The drawing library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Drivers.Sitronix.ST7735 par GHI Electronics, LLC	 v1.0.0
	Sitronix ST7735 driver for TinyCLR OS.	
	GHIElectronics.TinyCLR.Native par GHI Electronics, LLC	 v1.0.0
	The native library for TinyCLR OS.	
	GHIElectronics.TinyCLR.Pins par GHI Electronics, LLC	 v1.0.0
	The pins library for TinyCLR OS.	

Nugets à ajouter par un clic-droit sur Référence, "Gérer les packages NuGet"

4.2 Les cartes BrainPad

Il existe deux versions de la carte BrainPad. Selon la version, les possibilités de programmation et la procédure d'installation diffèrent.



Version 1 (obsolète) en 2018



Version 2

4.2.1 Procédures d'installation (Sept 2018)

	BrainPad version 1	BrainPad version 2	BrainPad version 2
Bootloader	V2.0.4	V1.0.2	V2.0.4
Firmware	TinyCLR OS V1.0.0 ou +	V10.0.0	TinyCLR OS V1.0.0 ou +
Programmation graphique avec MakeCode	Non	Oui	Non
Programmation en C# ou VB avec VS2017	Oui	Non	Oui

4.2.2 Mise en œuvre d'un programme sur la carte BrainPad V2 sous MakeCode

- Ecrire le programme avec MakeCode et le sauvegarder
- Transférer le fichier du programme dans la carte par glissé, déplacé en suivant la procédure ci-dessous
 - **BrainPad BP2 revA (proto)**
 - > Appuyer sur le bouton 'L' puis sur le bouton 'Reset' pendant 3s
 - > Relâcher 'Reset' puis relâcher 'L' => Light Bulb s'éclaire en vert pour indiquer que la carte est prête à recevoir le fichier au format .uf2 et une fenêtre de l'explorateur contenant un fichier INFO.TXT s'ouvre
 - > Faites glisser le fichier au format .uf2 dans cette fenêtre. Le Light Bulb clignote puis s'éteint. Le programme est prêt à fonctionner
 - **BrainPad BP2 revC ou sup (commercial)**
 - Même procédure sauf qu'il faut seulement appuyer sur **RESET jusqu'à ce que le Light Bulb devienne vert**

4.2.3 La carte BRAINPAD v1 (SoC G30)

4.2.3.1 GPIO - Commande d'un moteur pas à pas avec une carte Stepper click

Code C#

```
using System;
using System.Threading;
using testEasyStepperDriver;
using GTP = GHIElectronics.TinyCLR.Pins;

namespace HappyNewYear
{
    class Program
    {
        static bool toggle = false; UInt16 delay = 5; UInt32 nbpas = 30;
        EasyStepperDriver stepper;

        public void BrainPadSetup()
        {
            // Motor : SY57STH41-1006A (SYN=200 pas - U=5V - C=210g/cm) - Driver : STEPPER click
            // STEPPER click on mikroBUS connector
            stepper = new EasyStepperDriver(GTP.BrainPad.Expansion.GpioPin.Cs, GTP.BrainPad.Expansion.GpioPin.Pwm);

            BrainPad.Display.DrawText(0, 10, "Happy New"); BrainPad.Display.DrawText(0, 30, "Year 2018 !");
            BrainPad.Display.DrawSmallText(20, 55, "TinyCLR OS v0.7.0"); BrainPad.Display.ShowOnScreen();

            BrainPad.Buttons.WhenLeftButtonPressed += Buttons_WhenLeftButtonPressed;
            BrainPad.Buttons.WhenRightButtonPressed += Buttons_WhenRightButtonPressed;
        }

        private void Buttons_WhenRightButtonPressed()
        {
            toggle = false;
        }

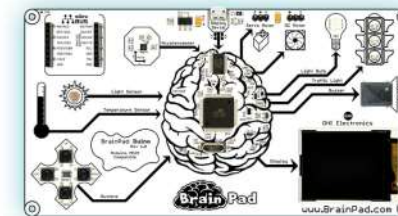
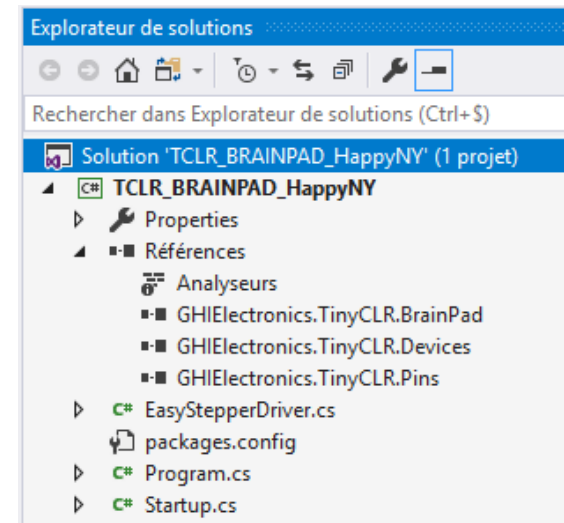
        private void Buttons_WhenLeftButtonPressed()
        {
            toggle = true;
        }

        public void BrainPadLoop()
        {
            stepper.StepMode = EasyStepperDriver.Mode.Full; stepper.StepDirection = EasyStepperDriver.Direction.Backward;

            if (toggle)
            {
                stepper.Turn(nbpas, EasyStepperDriver.Direction.Backward, delay); Thread.Sleep(1000);
                stepper.Turn(nbpas, EasyStepperDriver.Direction.Forward, delay); Thread.Sleep(1000);
            }
            Thread.Sleep(10);
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")



<https://goo.gl/iWHZGR>

4.2.4 La carte BRAINPAD v2 (STM32F401RE)

4.2.4.1 GPIO – Faire clignoter la led "Light Bulb" de la carte

Code C#

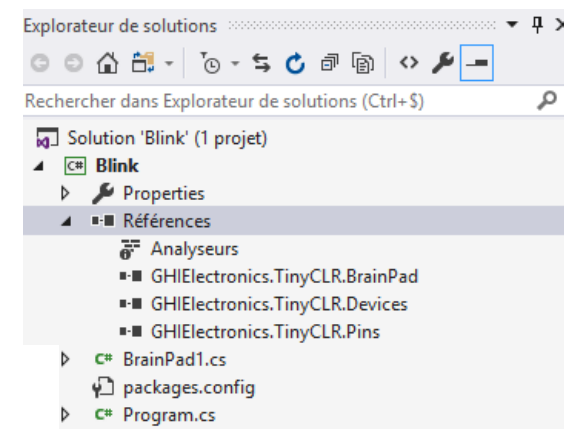
```
using GHIElectronics.TinyCLR.BrainPad;

namespace Blink
{
    class Program
    {
        static void Main()
        {
            BrainPad.Display.DrawText(0, 0, "Hello!");
            BrainPad.Display.RefreshScreen();

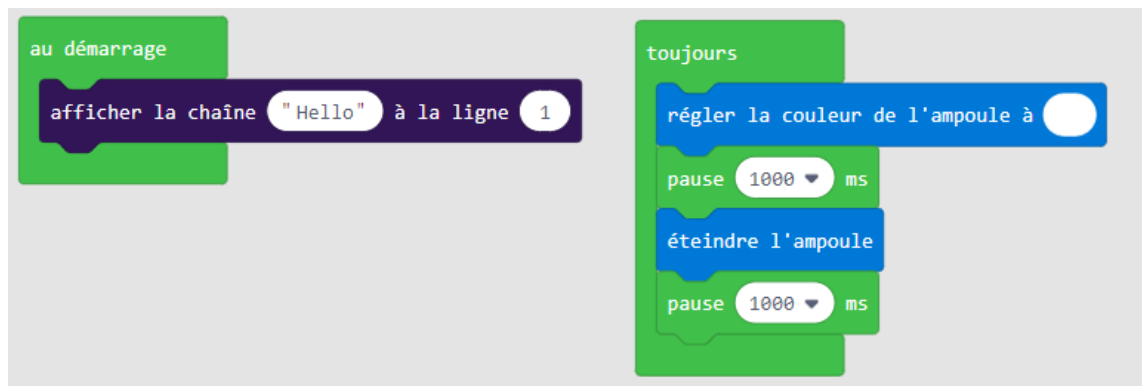
            while (true)
            {
                BrainPad.LightBulb.TurnWhite();
                BrainPad.Wait.Seconds(1);
                BrainPad.LightBulb.TurnOff();
                BrainPad.Wait.Seconds(1);
            }
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")



MakeCode



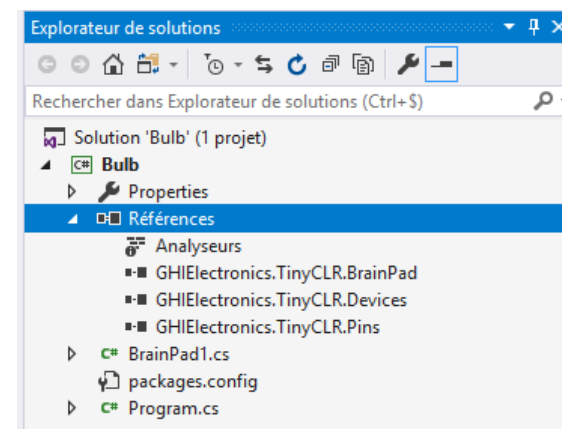
4.2.4.2 GPIO – Faire clignoter la led "Light Bulb" de la carte en bleu

Code C#

```
using GHIElectronics.TinyCLR.BrainPad;

namespace Bulb
{
    class Program
    {
        static void Main()
        {
            BrainPad.LightBulb.TurnOff();

            while (true)
            {
                BrainPad.LightBulb.TurnOff();
                BrainPad.Wait.Seconds(0.25);
                BrainPad.LightBulb.TurnBlue();
                BrainPad.Wait.Seconds(0.25);
            }
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")

MakeCode



4.2.4.3 GPIO – Réglage de la couleur de la led "Light Bulb" avec les boutons-poussoir

Code C#

```

using GHIElectronics.TinyCLR.BrainPad;

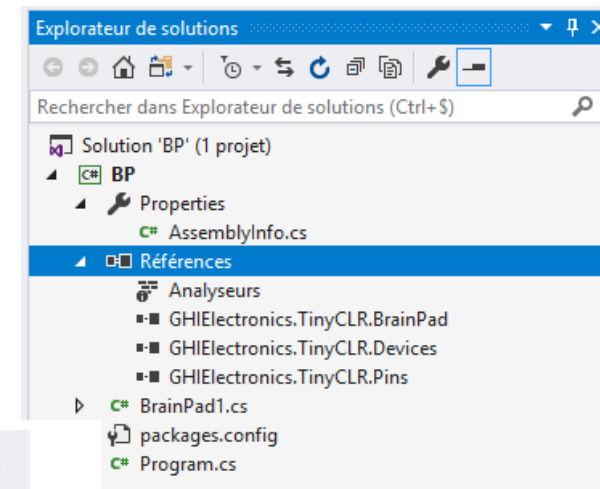
namespace BP
{
    class Program
    {
        static void Main()
        {
            BrainPad.LightBulb.TurnOff();

            while (true)
            {
                if (BrainPad.Buttons.IsDownPressed())
                    BrainPad.LightBulb.TurnGreen();
                else if (BrainPad.Buttons.IsLeftPressed())
                    BrainPad.LightBulb.TurnBlue();
                else if (BrainPad.Buttons.IsUpPressed())
                    BrainPad.LightBulb.TurnWhite();
                else if (BrainPad.Buttons.IsRightPressed())
                    BrainPad.LightBulb.TurnRed();
                else
                    BrainPad.LightBulb.TurnOff();
            }
        }
    }
}

```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")



MakeCode

```

au démarrage
    éteindre l'ampoule

toujours
    si bouton du bas ▼ est appuyé alors
        régler la couleur de l'ampoule à [vert]
    sinon
        si bouton de gauche ▼ est appuyé alors
            régler la couleur de l'ampoule à [bleu]
        sinon
            si bouton du haut ▼ est appuyé alors
                régler la couleur de l'ampoule à [blanc]
            sinon
                si bouton de droite ▼ est appuyé alors
                    régler la couleur de l'ampoule à [rouge]
                sinon
                    éteindre l'ampoule
    +
    +
    +
    +
    
```


4.2.4.4 Jouer "Au clair de la lune" avec le Buzzer V1a

Code C#

```
using GHIElectronics.TinyCLR.BrainPad;

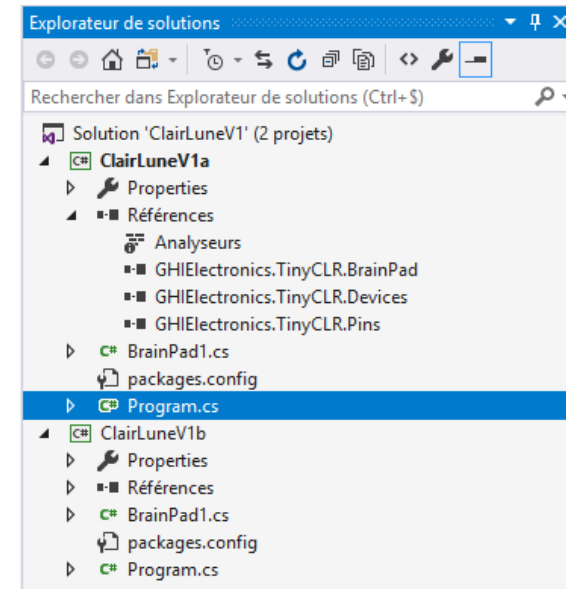
namespace ClairLuneV1a
{
    class Program
    {
        static void joueNote(int note, double tempo)
        {
            BrainPad.Buzzer.StartBuzzing(note);
            BrainPad.Wait.Seconds(tempo);
            BrainPad.Buzzer.StopBuzzing();
        }

        static void Main()
        {
            BrainPad.Display.DrawText(20, 5, "Au clair ");
            BrainPad.Display.DrawText(30, 25, "de la");
            BrainPad.Display.DrawText(20, 50, "lune v1a");
            BrainPad.Display.RefreshScreen();

            for (int i = 0; i < 3; i++)
            {
                joueNote(523, 0.5); // Do
            }

            joueNote(587, 0.5); // Ré
            joueNote(659, 1); // Mi
            joueNote(587, 1); // Ré
            joueNote(523, 0.5); // Do
            joueNote(659, 0.5); // Mi

            for (int i = 0; i < 2; i++)
            {
                joueNote(587, 0.5); // Ré
            }
            joueNote(523, 1.5); // Do
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")

4.2.4.5 Jouer "Au clair de la lune" avec le Buzzer V1b

Code C#

```
using GHIElectronics.TinyCLR.BrainPad;

namespace ClairLuneV1b
{
    class Program
    {
        static void joueNote(int note, double tempo)
        {
            BrainPad.Buzzer.StartBuzzing(note);
            BrainPad.Wait.Seconds(tempo);
            BrainPad.Buzzer.StopBuzzing();
        }

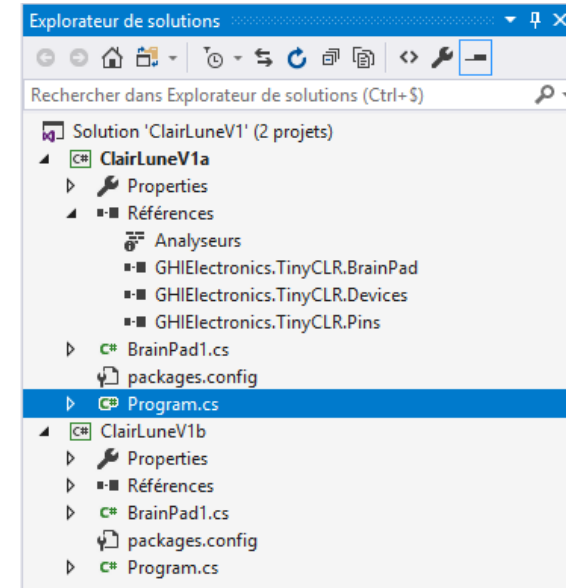
        static void Main()
        {
            BrainPad.Display.DrawText(20, 5, "Au clair ");
            BrainPad.Display.DrawText(30, 20, "de la");
            BrainPad.Display.DrawText(20, 35, "lune v1b");
            BrainPad.Display.DrawSmallText(30, 55, "Appuyer sur L");
            BrainPad.Display.RefreshScreen();

            while (!BrainPad.Buttons.IsLeftPressed()) ;

            BrainPad.Wait.Seconds(1);
            BrainPad.Display.Clear();

            for (int i = 0; i < 3; i++)
            {
                switch (i)
                {
                    case 0: BrainPad.Display.DrawSmallText(10, 10, "Au"); break;
                    case 1: BrainPad.Display.DrawSmallText(10, 10, "Au clair"); break;
                    case 2: BrainPad.Display.DrawSmallText(10, 10, "Au clair de"); break;
                    default:
                        break;
                }
                BrainPad.Display.RefreshScreen();
                joueNote(523, 0.5); // Do
            }
            BrainPad.Display.DrawSmallText(10, 10, "Au clair de la");
            BrainPad.Display.RefreshScreen();

            joueNote(587, 0.5); // Ré
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")

4.2.4.6 Jouer "Au clair de la lune" avec le Buzzer V2

Code C#

```

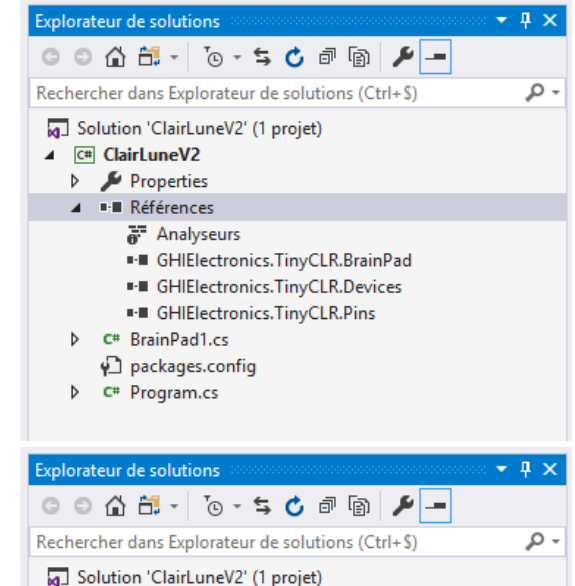
using System;
using GHIElectronics.TinyCLR.BrainPad;
using System.Threading;

namespace ClairLuneV2
{
    class Program
    {
        static Int32 posx = 0, posy = 0;
        const Int32 increment = 15;

        static bool debounce()
        {
            BrainPad.Wait.Milliseconds(30);
            if (BrainPad.Buttons.IsLeftPressed())
            {
                BrainPad.Buzzer.StartBuzzing(523); return true; //do
            }
            else if (BrainPad.Buttons.IsUpPressed())
            {
                BrainPad.Buzzer.StartBuzzing(587); return true; // ré
            }
            else if (BrainPad.Buttons.IsRightPressed())
            {
                BrainPad.Buzzer.StartBuzzing(659); return true; // mi
            }
            else
            {
                BrainPad.Buzzer.StopBuzzing(); return false;
            }
        }

        static void play(string note)
        {
            if (debounce())
            {
                BrainPad.Display.DrawSmallText(posx, posy, note);
                BrainPad.Display.RefreshScreen();
                posx += increment;
                if (posx > 119)
                {
                    posy += 15; posx = 0;
                }
            }
        }
    }
}

```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")

```

static void Main()
{
    BrainPad.Display.DrawText(20, 5, "Au clair ");
    BrainPad.Display.DrawText(30, 25, "de la");
    BrainPad.Display.DrawText(20, 50, "lune v2");
    BrainPad.Display.RefreshScreen();
    Thread.Sleep(2000);
    BrainPad.Display.Clear();

    BrainPad.Buttons.WhenLeftButtonPressed += Buttons_WhenLeftButtonPressed;
    BrainPad.Buttons.WhenLeftButtonReleased += Buttons_WhenLeftButtonReleased;
    BrainPad.Buttons.WhenUpButtonPressed += Buttons_WhenUpButtonPressed;
    BrainPad.Buttons.WhenUpButtonReleased += Buttons_WhenUpButtonReleased;
    BrainPad.Buttons.WhenRightButtonPressed += Buttons_WhenRightButtonPressed;
    BrainPad.Buttons.WhenRightButtonReleased += Buttons_WhenRightButtonReleased;

    Thread.Sleep(-1);
}

private static void Buttons_WhenRightButtonReleased()
{
    debounce();
}
private static void Buttons_WhenUpButtonReleased()
{
    debounce();
}
private static void Buttons_WhenLeftButtonReleased()
{
    debounce();
}
private static void Buttons_WhenLeftButtonPressed()
{
    play("do");
}
private static void Buttons_WhenUpButtonPressed()
{
    play("re");
}
private static void Buttons_WhenRightButtonPressed()
{
    play("mi");
}
}
}

```

4.3 La carte NETDUINO 3

4.3.1 GPIO - Blink

Source : <http://docs.ghielectronics.com/tinyclr/tutorials/gpio.html>

```
using System;
using System.Threading;
using GHIElectronics.TinyCLR.Devices.Gpio;

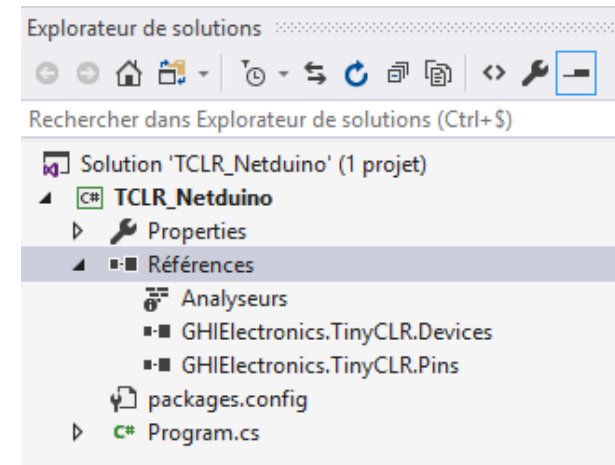
namespace TCLR_Netduino3_Blink
{
    class Program
    {
        static int PinNumber(char port, byte pin)
        {
            if (port < 'A' || port > 'E')
                throw new ArgumentException();

            return ((port - 'A') * 16) + pin;
        }

        static void Main()
        {
            var gpio = GpioController.Default;

            GpioPin led = gpio.OpenPin(PinNumber('A',10));
            led.SetDriveMode(GpioPinDriveMode.Output);

            while (true)
            {
                led.Write(GpioPinValue.High);
                Thread.Sleep(1000);
                led.Write(GpioPinValue.Low);
                Thread.Sleep(1000);
            }
        }
    }
}
```



Nugets installés (clic-droit sur Référence, "Gérer les packages NuGet")

A faire : classe pour les entrées – sorties de la carte Netduino3

Annexe 1 : L'outil TinyCLR Config

TinyCLR Config est un outil de gestion des cartes de développement chargées avec **TinyCLR OS**. Il est connecté au **cloud**. TinyCLR Config trouve et télécharge automatiquement les versions les plus récentes du firmware à partir d'une liste de sources fournies. Ces sources peuvent être sur Internet ou locales sur votre machine si vous n'avez pas ou ne voulez pas accéder à Internet. La fonctionnalité de mise à jour du micrologiciel ne fonctionne que pour les appareils équipés de GHI **Bootloader v2**.

- a) **Télécharger** la dernière version du logiciel sur le site de **GHI Electronics** et l'installer par un double clic sur le fichier .msi.

<https://goo.gl/Pmxasc>

TinyCLR Config

TinyCLR Config is a tool used to update and configure your TinyCLR device.

File	Date	Status	MD5
v0.7.0	2018-01-04	Alpha	329D4A24BA66423DD5D655202873B38C
v0.6.0	2017-08-31	Alpha	75743E33D1B98E6999BDCC9936479C14

- b) Lancer le logiciel  situé dans GHI Electronics.

- c) Sélectionner la carte et se connecter



Lorsque TinyCLR Config démarre, il recherche un fichier appelé **FirmwareSources.txt** dans le même répertoire que l'exécutable TinyCLR Config lui-même. Si le fichier n'est pas présent, il est créé automatiquement avec une entrée pointant vers le magasin de firmware GHI officiel.

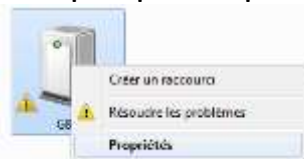
Le fichier est une liste de sources séparées par des virgules, une par ligne. La première valeur est une priorité entière avec les nombres inférieurs recherchés en premier. Chaque source est essayée jusqu'à ce qu'une version plus récente que la version actuellement flashée soit trouvée. La valeur suivante est Local ou Réseau qui détermine où pointe la valeur suivante : le disque local ou le réseau. La dernière valeur est un URI qui pointe vers un fichier xml qui définit la liste des firmwares disponibles.

La liste du firmware xml est composée d'une liste de périphériques. Chaque périphérique a un attribut de nom et un enfant de liste de micrologiciels. Dans la liste du firmware est une liste de chaque firmware disponible. Chaque micrologiciel doit avoir un hachage MD5 correspondant au fichier pointé, une version du formulaire major.minor.patch, une date de publication et un uri qui pointe vers l'emplacement du micrologiciel (le type d'emplacement doit correspondre au type d'emplacement de la liste de firmware elle-même, c'est-à-dire locale vs réseau). Voir le fichier hébergé officiel pour un exemple complet.

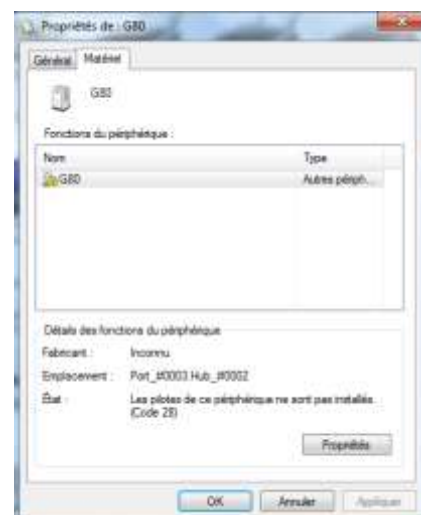
Annexe 2 : Résolution du problème d'installation du driver de la carte sous Windows 7

Suivre la procédure ci-dessous si la carte n'est pas reconnue par Windows 7.

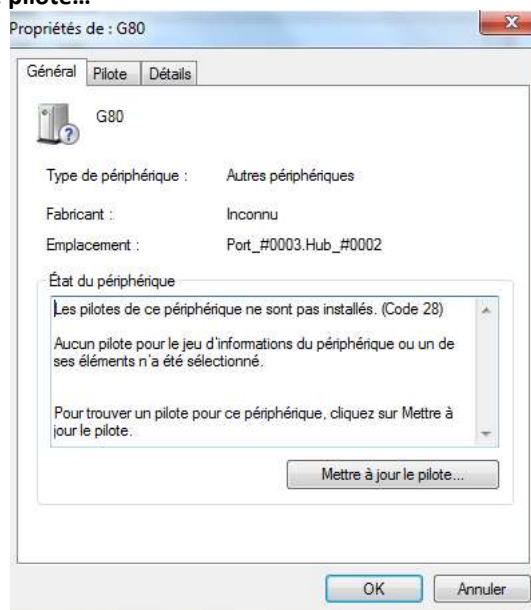
- a) Clic droit sur l'icône du périphérique dans "**Périphériques et imprimantes**".



- b) Sélectionner l'onglet "**Matériel**" puis "**Propriétés**" dans la fenêtre ci-dessous.



- c) Sélectionner "**Mettre à jour le pilote...**"



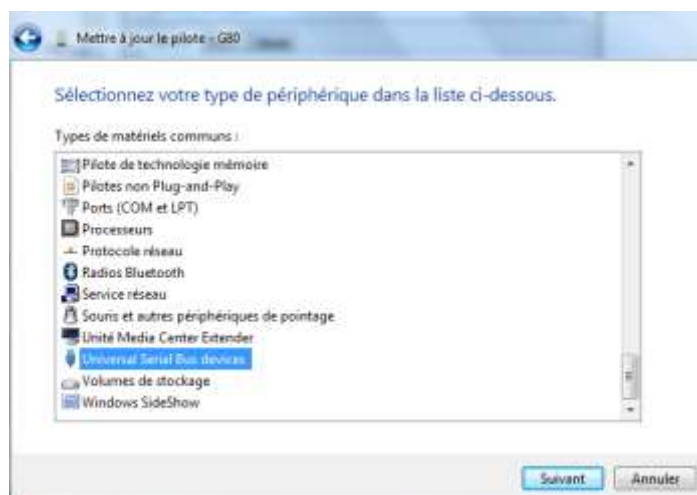
- d) Clic sur l'encadré ci-dessous dans la fenêtre.

➔ Rechercher un pilote sur mon ordinateur
 Recherchez et installez manuellement le pilote.

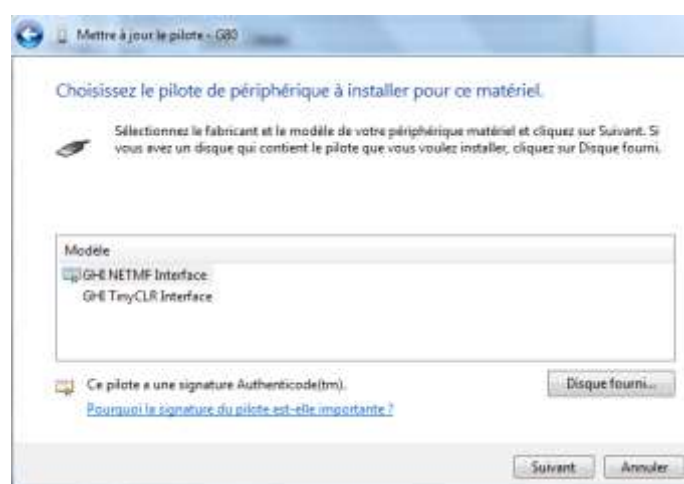
- e) Sélectionner l'encadré ci-dessous

➔ Choisir parmi une liste de pilotes de périphériques sur mon ordinateur
 Cette liste affichera les pilotes installés et compatibles avec le périphérique, ainsi que tous les pilotes dans la même catégorie que le périphérique.

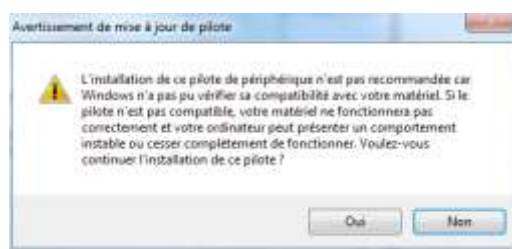
- f) Sélectionner "**Universal Serial Bus devices**" puis clic sur "**suivant**"



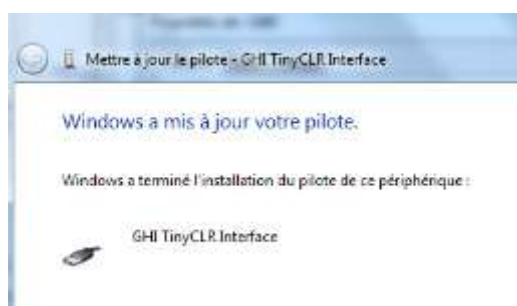
g) Sélectionner le pilote correspondant à la carte puis "Suivant"



h) Clic sur "Oui"



i) La fenêtre ci-dessous indique que l'installation a été correctement réalisée.



Le pilote a maintenant l'apparence suivante

