

**TP n°2 : Windows 10 IoT Standard - CORRECTION****1. - Prise en main de Windows 10 IoT Standard :**

Allez sur <https://docs.microsoft.com/en-us/windows/iot-core/tutorials/quickstarter/PrototypeBoards>, quels types de cartes électroniques sont utilisables avec Windows 10 IoT Core ? :

*Raspberry PI2 et 3, DragonBoard 410C, Minnowboard Max, NXP i.MX, ...*

Connectez physiquement votre Raspberry à un écran, à un câble ethernet et à un clavier et à une souris.  
Connectez y votre carte SD et démarrez-la.

Démarrez votre système Windows IoT, il doit vous afficher l'adresse IP qu'il a récupéré en Dhcp.

Qu'affiche votre système ? :

*Une simple interface d'accueil*

Avez-vous une interface utilisateur et des outils interactifs ? :

*Non, juste des outils d'info et une invite de command*



...

Allez sur « Apps/Apps Manager ». Vous voyez une liste d'applications.

A quoi doit correspondre l'application « IOTCoreDefaultApplication » ? :

*A l'application de démarrage, le bureau donc*

Allez sur « Apps/File Explorer ». Vous avez accès aux répertoires accessibles aux applications.

Dans quel sous-répertoire sont-elles installées ? :

*User Folders \ LocalAppData \*

Allez dans « Processes/Détails ». Que voyez-vous ? :

*La liste des processus*

Et enfin allez dans « Processes/Performance », que voyez-vous ? :

*Les performances CPU, Entrées/Sorties, Mémoire et Réseau*

Allez dans « Processes /Run Command » pour accéder à l'interface PowerShell et tapez « ipconfig » et cliquez sur « Run ».

Tapez « dir », sur quel répertoire êtes-vous ? :

*C:\windows\system32*

...

**2. – Création d'une application avec Visual Studio :**

Laissez par défaut les versions proposées par Visual Studio (au minimum la version build 17763).

Quelle est la particularité des Applications Universelles dites « UWP » ? :

*Ce sont des applications qui peuvent s'exécuter sur tous les systèmes compatibles UWP (Windows 10/11)*

Il y a 2 fichiers relatifs à votre fenêtre principale. Quels sont-ils et quels types de code contiennent-ils ? :

*MainWindow.xaml => code Xaml de l'interface*

*MainWindow.xaml.cs => code C# associé à cette interface*

Ouvrez votre page « MainPage.xaml » et ajoutez un bouton au centre de votre fenêtre avec un « TextBox » juste en dessous. Quel langage est utilisé pour décrire l'interface ? :

*Xaml*

...

Que représente la balise `<Grid>` ? : *La grille de disposition de votre fenêtre*

## MainWindow.xaml

```
<Page
  x:Class="_2024_TP2.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:_2024_TP2"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="1*" />
      <RowDefinition Height="1*" />
    </Grid.RowDefinitions>
    <Button Content="Button" Grid.Row="0" Height="109" Margin="0,0,0,0"
VerticalAlignment="Top"
      Width="100" HorizontalAlignment="Left" Click="Button_Click"/>
    <TextBox x:Name="tb1" Grid.Row="1" HorizontalAlignment="Left" Margin="50,0,0,0"
Text="TextBox"
      TextWrapping="Wrap" VerticalAlignment="Top" Height="154" Width="400"/>
    <Button Content="WSMeteo" Grid.Row="0" Height="109" Margin="50,0,0,0"
VerticalAlignment="Top"
      Width="100" HorizontalAlignment="Center" Click="Button2_Click"/>
    <Button Content="Envoi" Grid.Row="0" Height="109" Margin="50,0,0,0"
VerticalAlignment="Top"
      Width="100" HorizontalAlignment="Right" Click="Button3_Click"/>
    <Slider x:Name="sd1" Margin="0,50,0,0" HorizontalAlignment="Center" Height="40"
Width="80" Maximum="20" ValueChanged="sd1_ValueChanged"/>
  </Grid>
</Page>
```

## MainWindow.xaml.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.Networking.Connectivity;
using Windows.Networking;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// Pour plus d'informations sur le modèle d'élément Page vierge, consultez la page
https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace _2024_TP2
{
    /// <summary>
    /// Une page vide peut être utilisée seule ou constituer une page de destination au sein d'un
    frame.
    /// </summary>
    public sealed partial class MainPage : Page
```

```

{
    public MainPage()
    {
        this.InitializeComponent();
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
        var deviceFamily = Windows.System.Profile.AnalyticsInfo.VersionInfo.DeviceFamily;
        tb1.Text = deviceFamily + Environment.NewLine;

        tb1.Text += "IP: " + GetFirstLocalIp();
    }

    public static string GetFirstLocalIp(HostNameType hostNameType = HostNameType.Ipv4)
    {
        var icp = NetworkInformation.GetInternetConnectionProfile();

        if (icp?.NetworkAdapter == null) return null;
        var hostname =
            NetworkInformation.GetHostNames()
                .FirstOrDefault(
                    hn =>
                        hn.Type == hostNameType &&
                        hn.IPInformation?.NetworkAdapter != null &&
                        hn.IPInformation.NetworkAdapter.NetworkAdapterId ==
                            icp.NetworkAdapter.NetworkAdapterId);
        return hostname?.CanonicalName;
    }

    private async void Button2_Click(object sender, RoutedEventArgs e)
    {
        WSMeteo.Service1Client sr = new WSMeteo.Service1Client();
        var val = await sr.Get_MeteoDataAsync();

        tb1.Text = "Temp: " + val.d_Temp.ToString("0.00") + " °C" + Environment.NewLine;
        tb1.Text += "Pression: " + val.d_Pres.ToString("0.0") + " hPa" + Environment.NewLine;
    }

    private async void Button3_Click(object sender, RoutedEventArgs e)
    {
        String sVal = "##BCY_3IL##" + sd1.Value.ToString();
        WSIoT.Service1Client sw = new WSIoT.Service1Client();
        String rep = await sw.SendDataIoTAsync(sVal);
        if (rep == "1")
        {
            tb1.Text = "Valeur bien envoyée et reçue";
        }
    }

    private void sd1_ValueChanged(object sender, RangeBaseValueChangedEventArgs e)
    {
        tb1.Text = "Valeur Slider : " + sd1.Value.ToString();
    }
}
}

```