


<p style="text-align: center;"><i>Système Numérique</i></p>	<p style="text-align: center;"><i>Interface graphique Visual C# Express Liaison série RS232</i></p>	BTS SN
		

1. Présentation : *Principe du projet et de la solution en Visual C#*

Visual C# est un IDE, c'est à dire un environnement de développement intégré (EDI en français).

Avant de pouvoir compiler quoique ce soit, il est nécessaire de créer un projet.

Un projet est un environnement de compilation qui décrit tout ce qu'il faut pour parvenir à compiler et construire un programme. Cela inclut bien sûr la liste des fichiers à compiler, mais aussi l'ensemble des options qu'il faut passer au compilateur pour y parvenir.

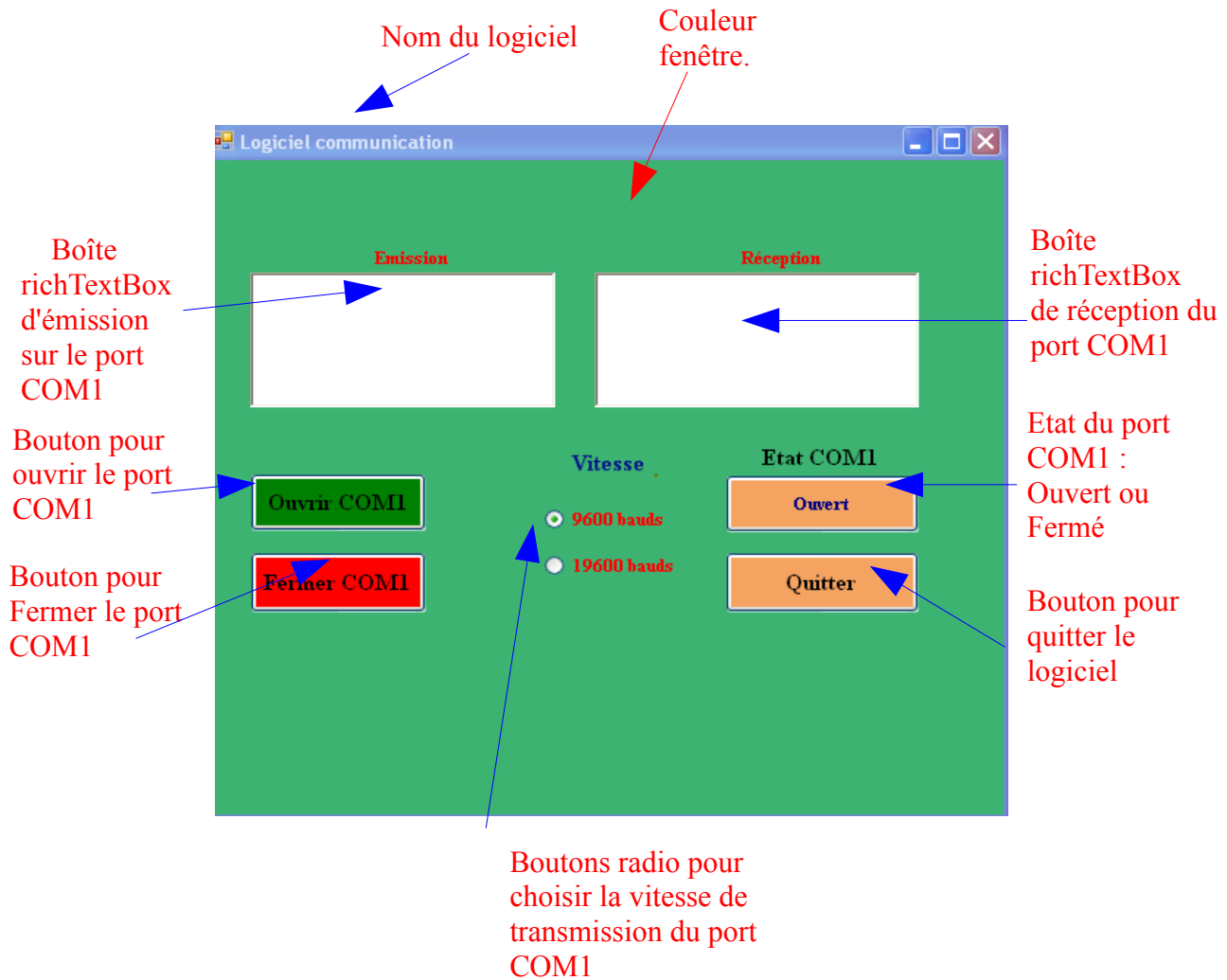
Visual C# Express permet en effet de cibler deux types d'application :

. **Application console** : il s'agit du développement textuel classique (fenêtre "invite de commandes). Les programmes sont écrits en C traditionnel.

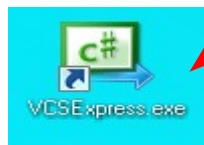
. **Application Windows Forms** : une programmation graphique utilisant l'API windows (fenêtre windows).

2. Création projet : *Logiciel communication liaison série*

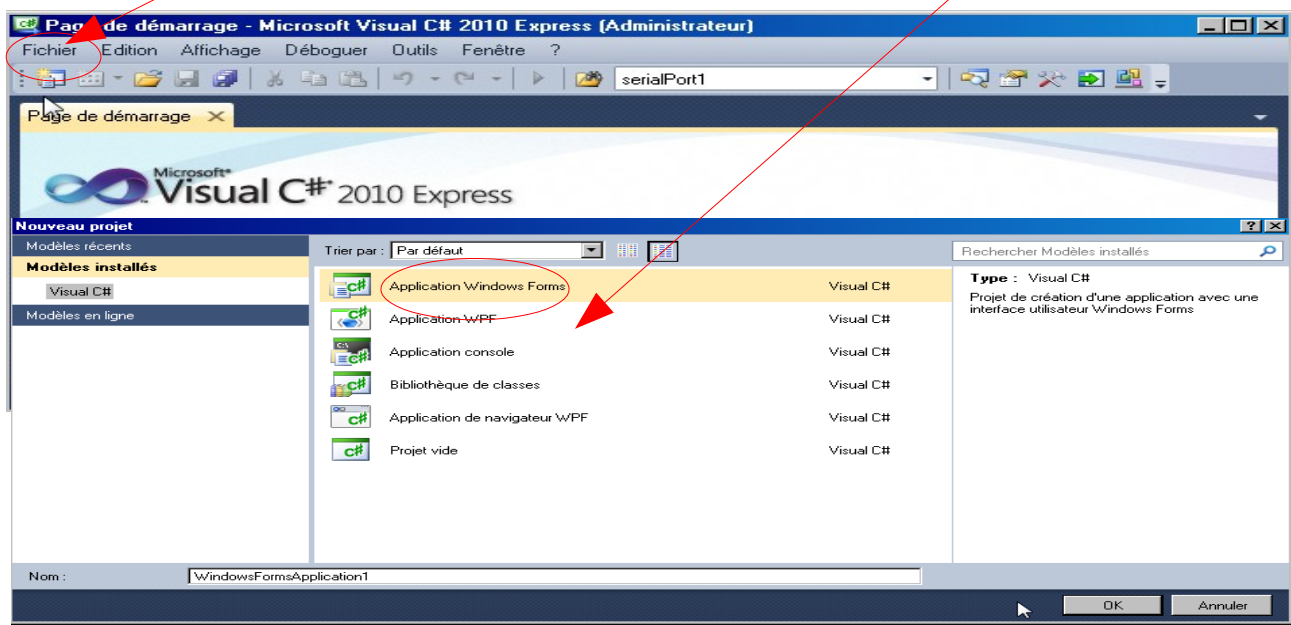
- On souhaite créer à l'aide de Visual C# Express un logiciel pour gérer la communication série du PC par l'intermédiaire des ports de communication (COM1, COM2,...) du PC. Ce logiciel aura l'interface graphique suivante :

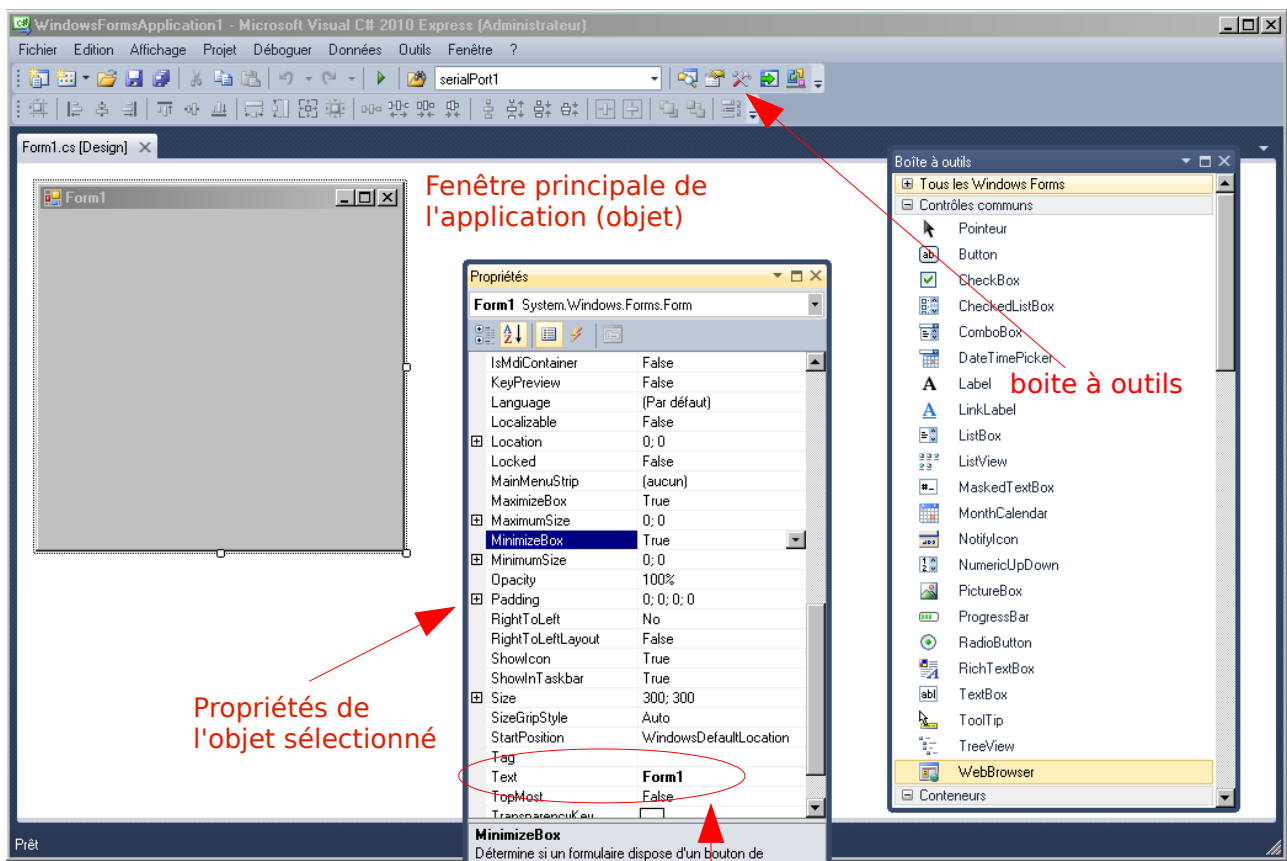


- Lancer le logiciel VISUAL C# en cliquant sur l'icône ou en passant par le menu Démarrer → Tous les programmes et choisir le logiciel.



3. Création d'un projet pour développer une application graphique en C# (donner un nom significatif)
- Cliquer sur fichier → Nouveau projet → Sélectionner Application Windows Forms





4. Placer les objets sur la fenêtre principale

Définir le titre de la fenêtre

- . Inscrire le titre dans la propriété "text" de la fenêtre Form1

Placer un bouton

- . Choisir l'objet **"button"** dans la boîte à outils
- . Placer le bouton dans la fenêtre "Form1"
- . Modifier le nom dans la propriété "text" de l'objet Button1

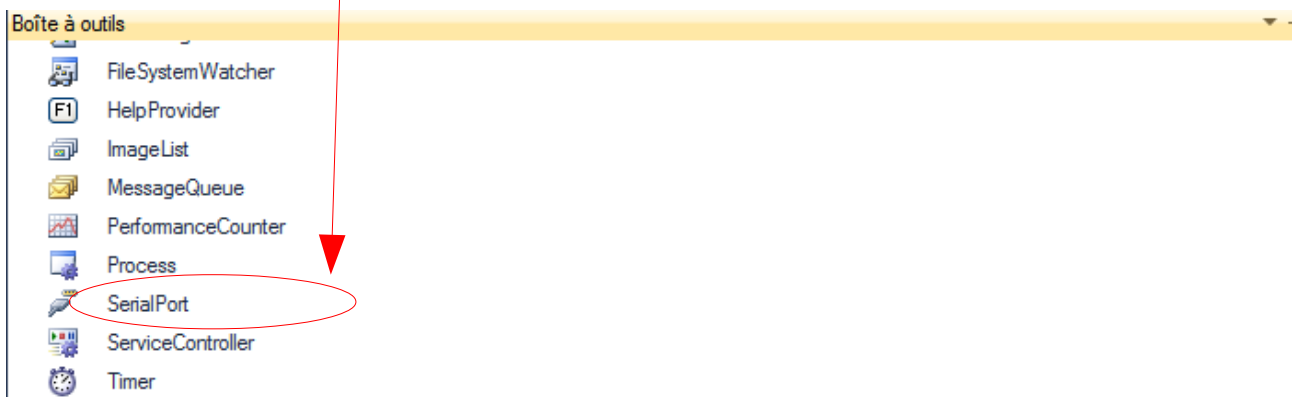
Placer un boîte richTextBox

- . Choisir l'objet **"richTextBox"** dans la boîte à outils
- . Placer la boîte richTextBox dans la fenêtre "Form1"
- . Modifier le nom dans la propriété "text" de l'objet richTextBox

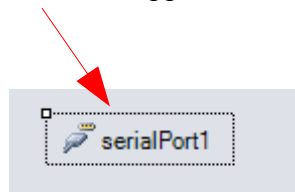
Placer un bouton radioButton

- . Choisir l'objet **"radioButton"** dans la boîte à outils
- . Placer le radioButton dans la fenêtre "Form1"
- . Modifier le nom dans la propriété "text" de l'objet radioButton

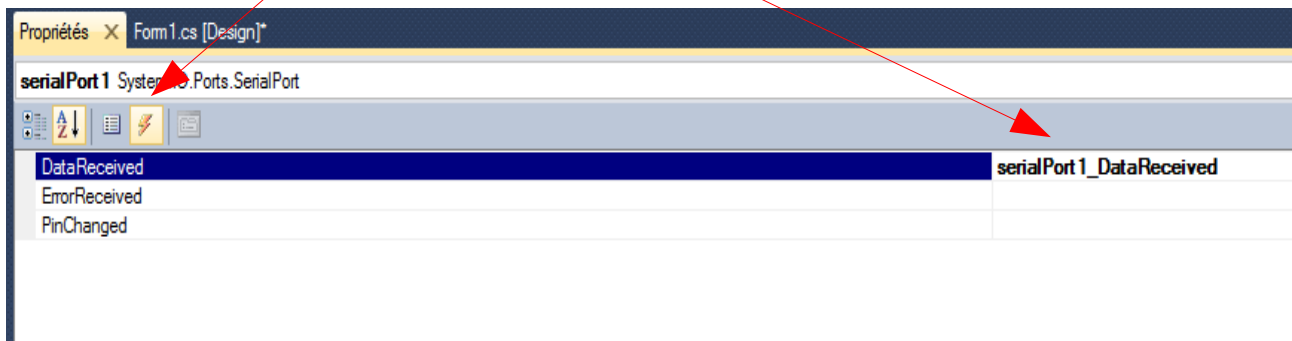
- Insérer un objet **"SerialPort1"** et définir les propriétés de cet objet (Port, vitesse, parité, bits de stop)



En bas de la fenêtre l'objet **SerialPort1** apparaîtra.



- Cliquer droit sur l'objet **SerialPort1** et sélectionner **Propriétés**.
- Sélectionner Événements et compléter la fenêtre



5. Définir l'action d'un bouton

- Double cliquer sur le bouton (ici button1), le code correspondant apparaît

```
private void button1_Click(object sender, EventArgs e)
{
    // insérer les lignes de programme effectuées par le bouton 1 ici
}
```

- Ajouter un bouton "Ouvrir COM1" et insérer le code pour ouvrir la communication série.

Exemple :

A chaque fois que je clique sur le bouton "Ouvrir COM1", je veux exécuter une action pour ouvrir

le port COM1. Pour cela il faut insérer le code suivant qui fait appel à la méthode

`serialPort1.Open(); // Ouverture du port COM1`

```
private void bouton1_Click(object sender, EventArgs e)
{
    serialPort1.Open();
}
```

Remarque très importante : si vous essayer d'ouvrir un port COM1 qui est déjà ouvert, une erreur sera générée. Il est donc conseillé de dévalider le bouton **Ouvrir COM1** avec l'instruction **bouton1.Enabled = false ;**

Pour valider le bouton ouvrir COM on utilise l'instruction **bouton1.Enabled = true ;**

Pour résumer : dès que l'on ouvre le port COM1, on valide le bouton **Fermer COM1** et on dévalide le bouton **Ouvrir COM1** et dès qu'on ferme le port COM1, on dévalide le bouton **Fermer COM1** et on valide le bouton **Ouvrir COM1**

- **Ajouter** un bouton "Fermer COM1", puis compléter le code pour déconnecter la communication série.

Pour fermer le port COM1 on utilise la méthode `serialPort1.Close();`

- Ajouter un bouton "Quitter" pour quitter le logiciel avec la méthode **Close()**.
- **Ajouter** les boutons radios pour choisir la vitesse de communication.

6. Envoi de données sur le port COM1

Pour envoyer **Bonjour + un retour chariot** sur la liaison série on utilise :
`serialPort1.WriteLine("Bonjour " + "\r");`

Pour écrire dans un label on utilise : `label1.Text = "Ouvert"`

Pour choisir la vitesse de 9600 bauds on utilise : `serialPort1.BaudRate = 9600;`

Pour lire les caractères qui arrivent sur le port COM1 dans une richTextBox1; on utilise :

`richTextBox1.Text= serialPort1.ReadLine(); // lecture caractères reçus`

Remarque : `serialPort1.ReadLine()` lit les caractères jusqu'à la valeur retour à la ligne dans la mémoire tampon d'entrée.

Pour envoyer les caractères de la boîte richTextBox2 sur le port COM1; on utilise :

`serialPort1.WriteLine(richTextBox2.Text+ "\r");`

Compilation et exécution

Une fois le fichier source créé, la compilation peut être lancée et un fichier exécutable (nom_projet.exe) peut être généré par la touche F6.

- Tester le fonctionnement de votre programme en envoyant sur le port COM1 le texte contenu dans la boîte **richTextBox** d'émission et relever à l'aide d'un oscilloscope les

caractères envoyés sur la liaison série.

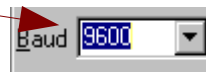
- Décoder les trames à l'aide d'un oscilloscope.
- Tester le fonctionnement de la partie réception dans la fenêtre de réception.

7. Amélioration : *Ajout de plusieurs fonctionnalités*

- Ajouter la fonctionnalité qui permet d'afficher l'état du port de communication (Ouvert ou fermé)
- Ajouter la fonctionnalité qui permet à l'utilisateur de choisir le port de communication dans une liste.



- Ajouter la fonctionnalité qui permet à l'utilisateur de choisir une vitesse de communication (600, 1200, , 9600 , ,115200,)



- Ajouter d'autres fonctionnalités qui paraissent importantes pour un logiciels de communication série (voir logiciel terminal, hyper terminal,.....).