Nom:…… Prénom:……..

TP SIN
Etude liaison série
Détailler tous les calculs et expliquer les réponses

Pré requis (l'élève doit savoir):
Savoir utiliser un ordinateur
Réaliser un programme sur C++ Builder
Réaliser un programme sur Arduino

Programme

Objectif terminal :
L'élève doit être capable de récupérer ou d'envoyer des informations sur une liaison série

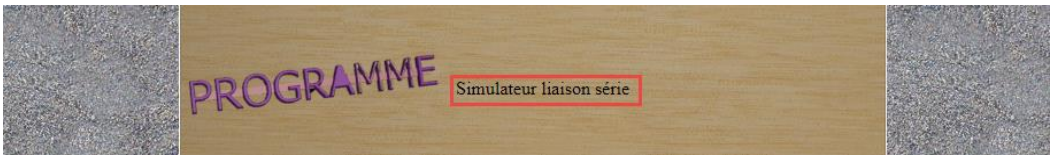
Matériel

- Ordinateur
- logiciel simulateur de trame série
- carte Arduino méga
-leds
-boutons
-plugin Tcomport de Winsoft http://www.winsoft.sk/comport.htm

1. **Travail demandé**
   a. Etude liaison série RS232

- Ouvrir le logiciel simulateur de trame liaison série

http://sti2dsinhyrome.fr/doc%20cours/liaison/serie/docserie.html

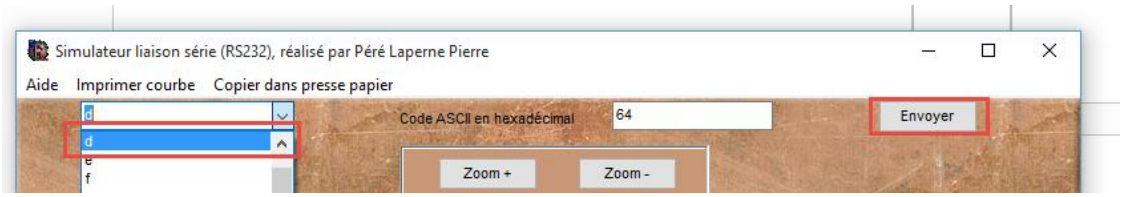


- Regarder la vidéo de fonctionnement

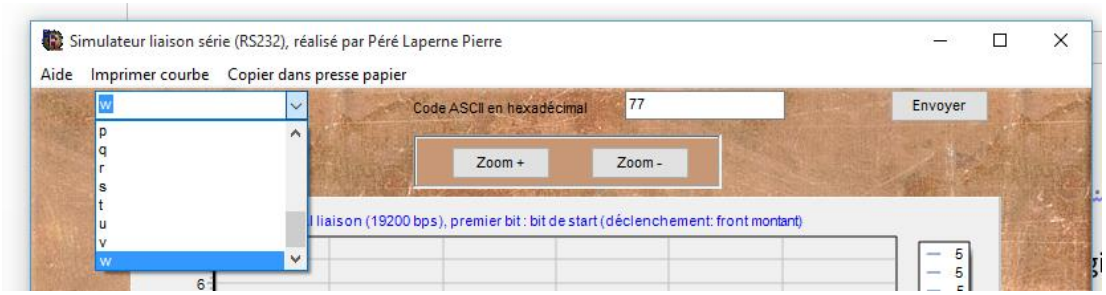
http://sti2dsinhyrome.fr/tp%20sin%20ressource%20video5.html

Nom:…… Prénom:……..

- Choisir la lettre d et envoyer la trame



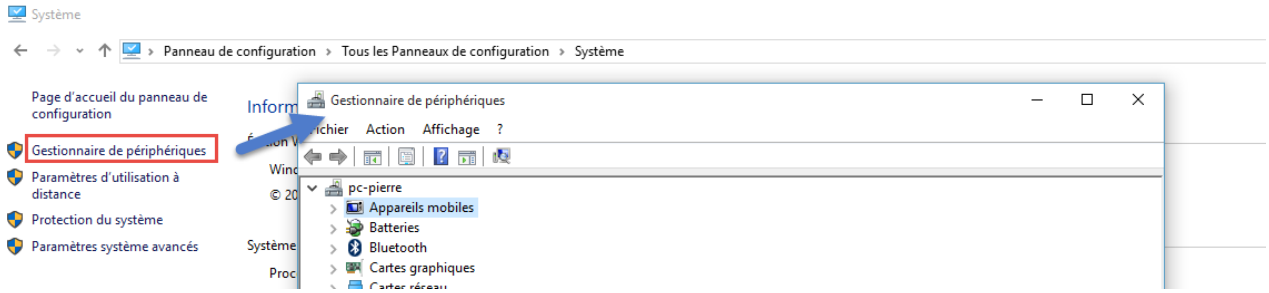
- Imprimer la courbe et coller là sur votre TP
- Indiquer sur la courbe le bit de Start, les 8 bits de données, le bit de parité et le bit de stop.
- Donner le bit de parité, et donner la valeur en binaire de la donnée envoyée
- Calculer la valeur en hexa et comparer là à la valeur du logiciel
- Choisir la lettre w et envoyer la trame



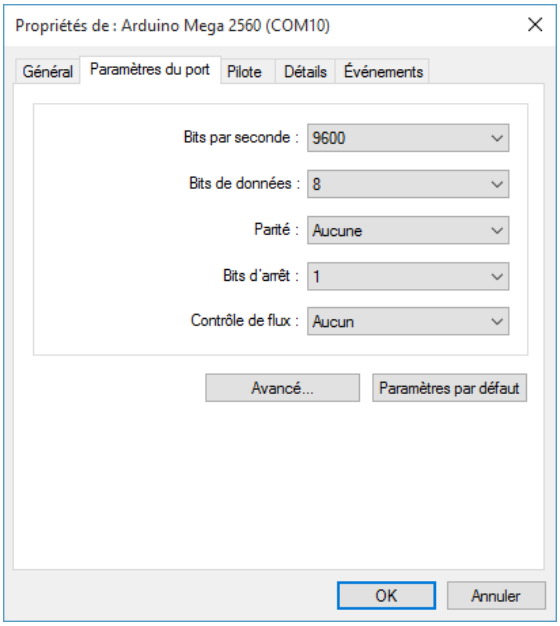
- Imprimer la courbe et coller là sur votre TP
- Indiquer sur la courbe le bit de Start, les 8 bits de données, le bit de parité et le bit de stop.
- Donner le bit de parité, et donner la valeur en binaire de la donnée envoyée
- Calculer la valeur en hexa et comparer là à la valeur du logiciel
- D'après les deux trames relevées, expliquer le fonctionnement du bit de parité.

b. Programmation liaison série

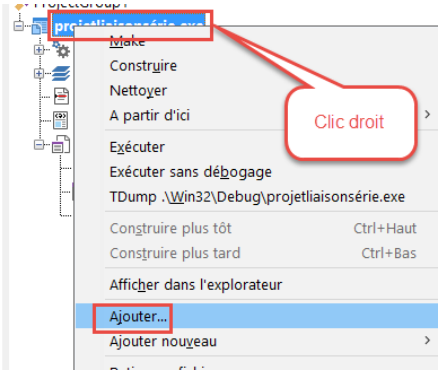
- Brancher la carte Arduino et indiquer les caractéristiques du port COM (numéro du port, vitesse de transmission (baud), parité etc..)

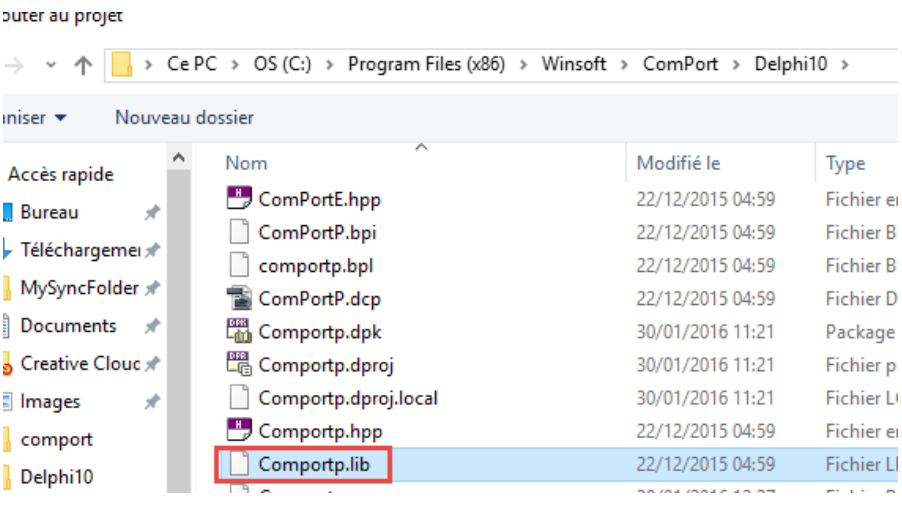


Nom:…… Prénom:……



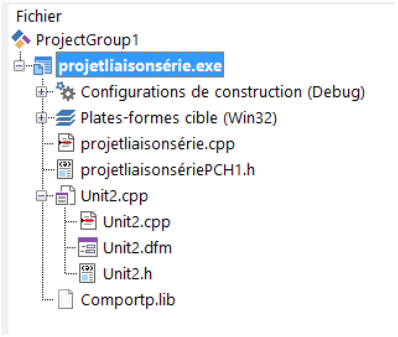
- Créer un nouveau projet VCL C++ Builder
- Enregistrer la librairie Comportp.lib dans votre projet



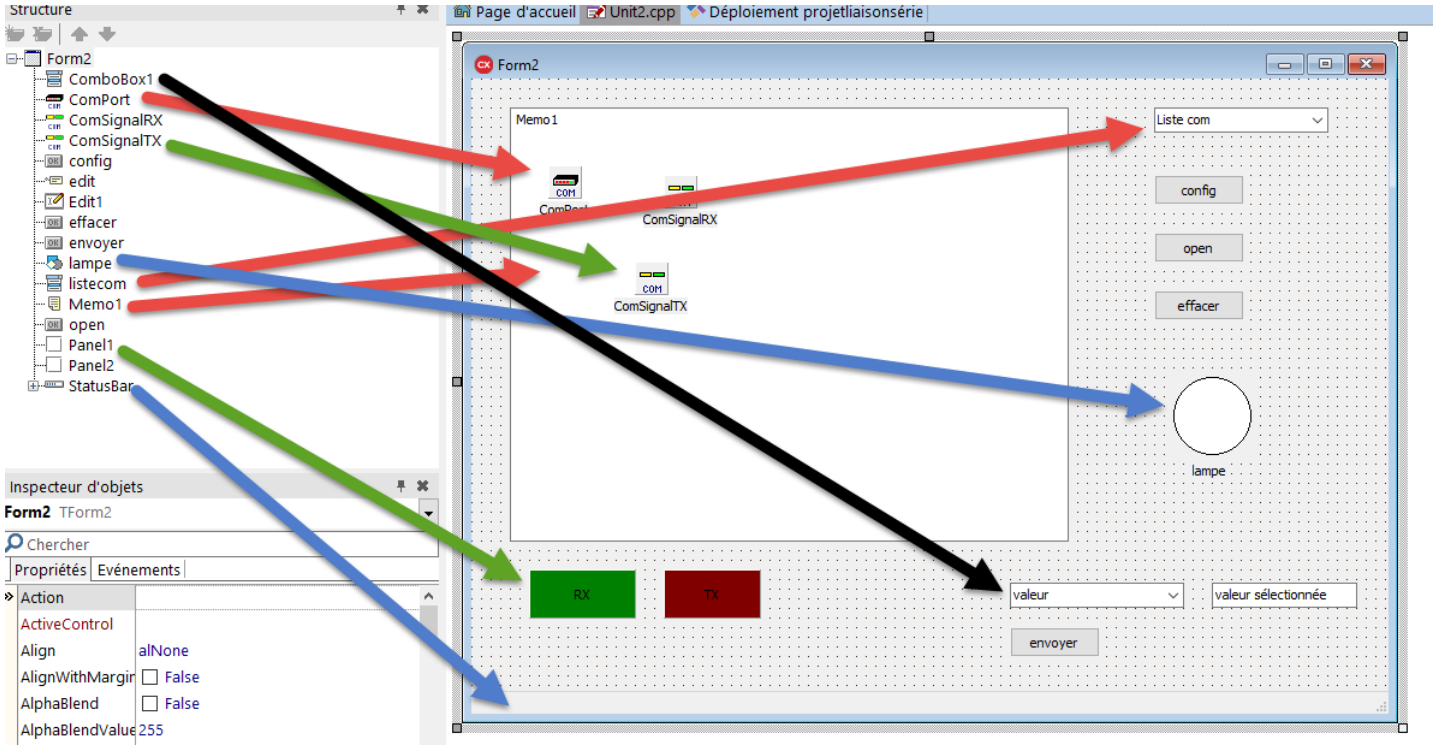


Nom:…… Prénom:……..

- Après avoir réouvert le programme, vous devez obtenir ce document



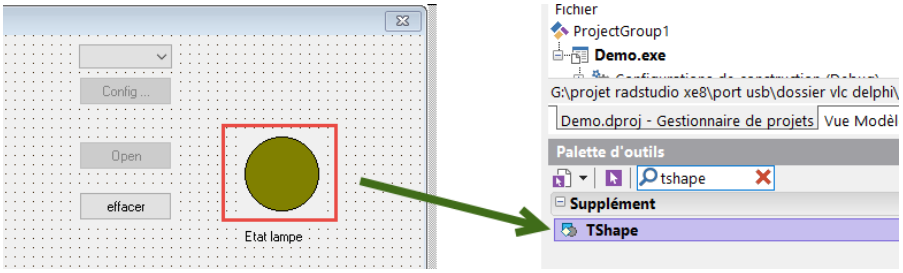
- Réaliser la page suivante

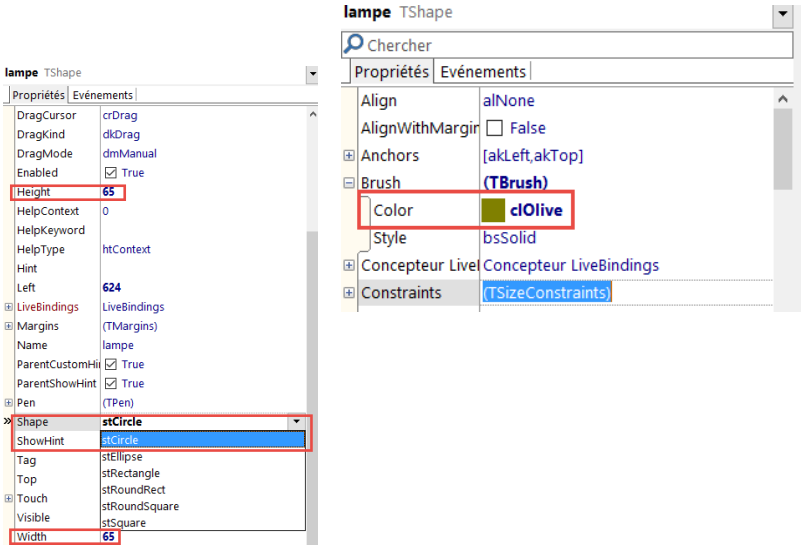


Broche	E/S	Désignation	Description
1	Entrée	DCD (Data Carrier Detect)	Détection de porteuse.
2	Entrée	RXD ou RD (Receive Data)	Réception de données.
3	Sortie	TXD ou TD (Request Data)	Emission de données.
4	Sortie	DTR (Data Terminal Ready)	Terminal prêt.
5	-	GND (Ground)	Masse.
6	Entrée	DSR (Data Set Ready)	Emmission prête.
7	Sortie	RTS (Request To Send)	Demande d'emmission.
8	Entrée	CTS (Clear To Send)	Prêt à emettre.
9	Entrée	RI (Ring Indicator)	Indicateur de sonnerie.

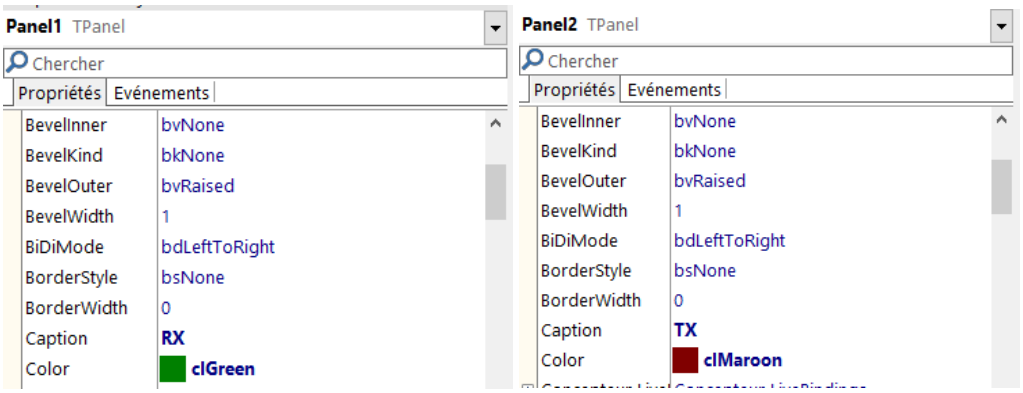
- Configurer la lampe suivant les paramètres ci-dessous

Nom:…… Prénom:……

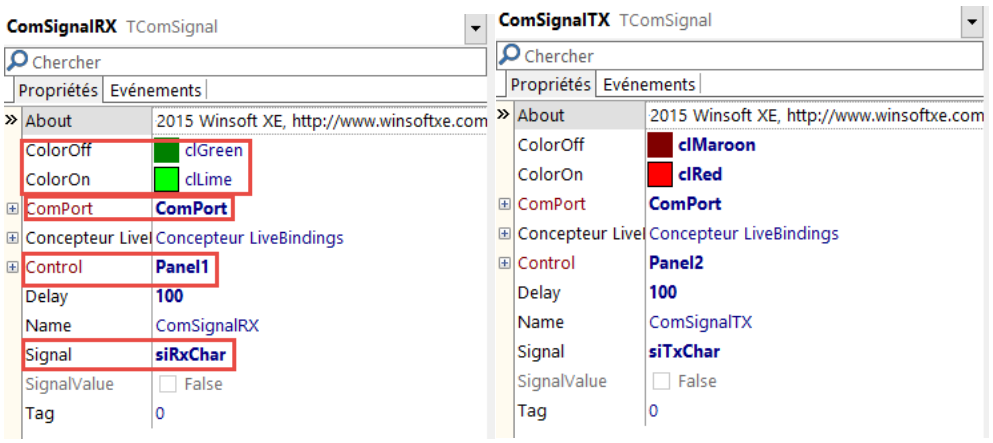




- Paramétrer les « panels » suivant :

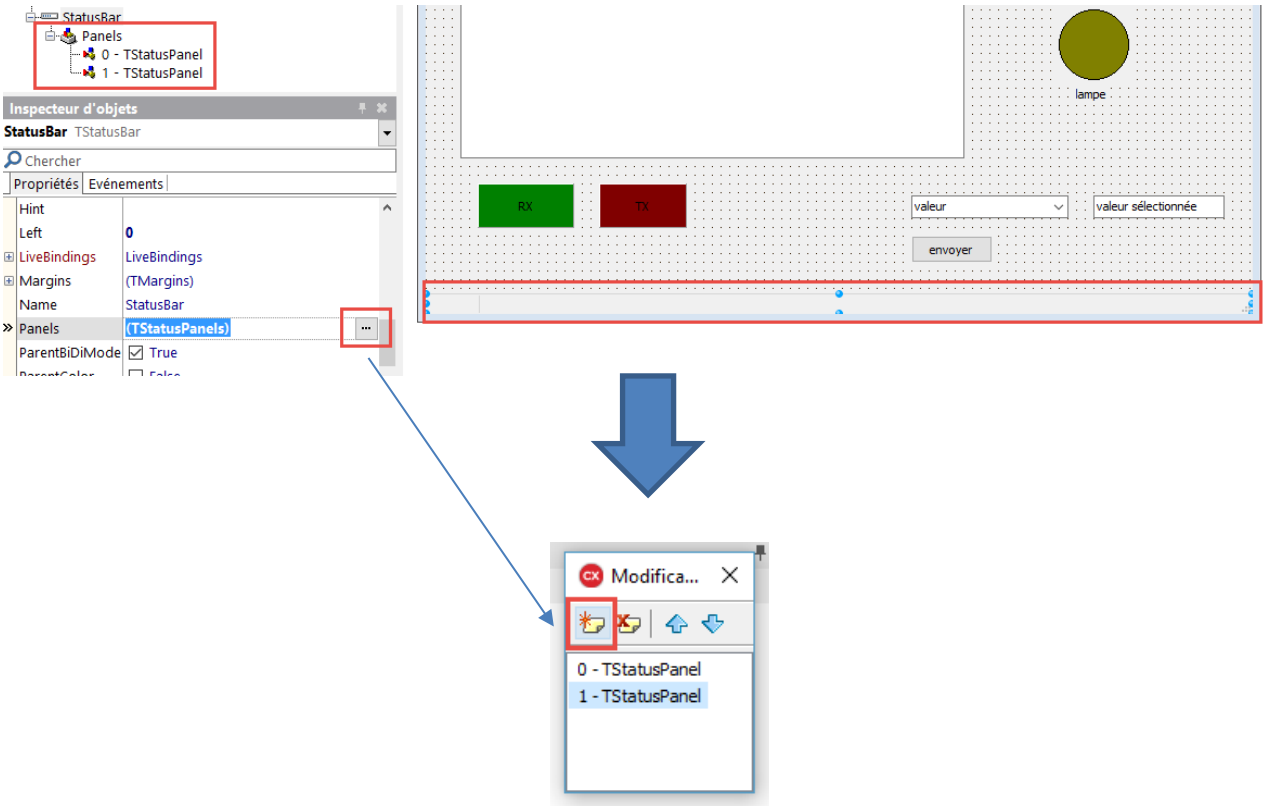


- Paramétrer les « comsignal » suivant :

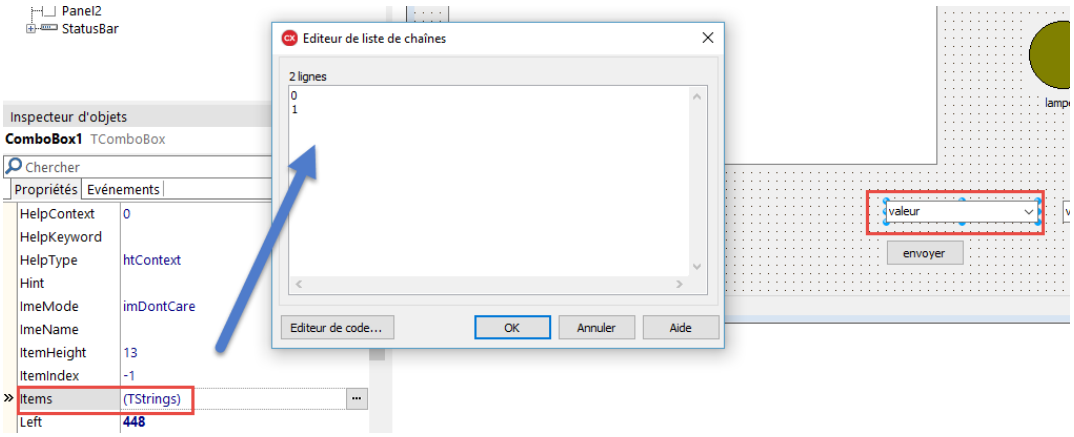


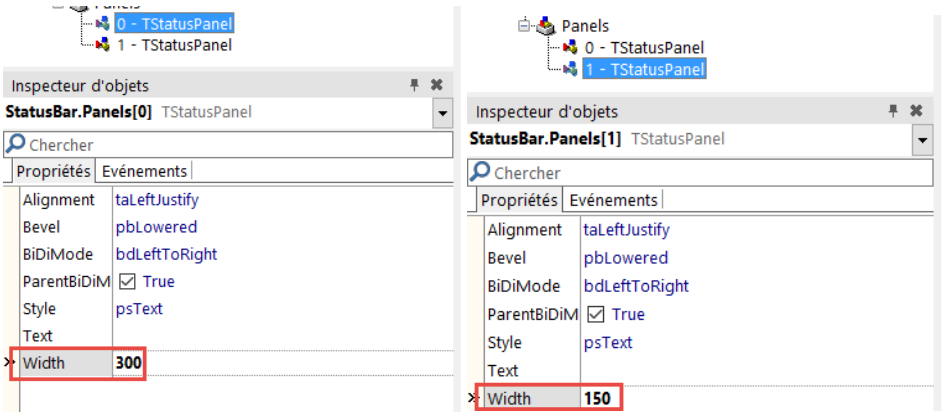
Nom:…… Prénom:……

- Paramétrer la Statusbar pour rajouter deux statuspanel



- Rajouter dans le combobox les deux valeurs 0 et 1





Nom:…… Prénom:……..

- Dans le programme, rajouter les lignes suivantes pour que la lampe à l'ouverture du programme soit de couleur olive et qu'on puisse récupérer la liste des ports com. De plus on ne doit pas pouvoir actionner les boutons si aucun port com est sélectionné.

```
void __fastcall TForm2::FormCreate(TObject *Sender)
{
this->lampe->Brush->Color=clOlive;
this->config->Enabled=false;
   this->open->Enabled=false;
    this->envoyer->Enabled=false;
ComPort->EnumComDevicesFromRegistry(listecom->Items);   //on récupère la liste des ports com dans listecom
if (this->listecom->Items->Count>0) {   //on teste si la liste est pleine
    listecom->ItemIndex = 0;
  ComPort->DeviceName = "\\\\.\\" + listecom->Text;
  this->config->Enabled=true;
   this->open->Enabled=true;

}

  AddReadBytes(0);     //permet de compter le nombre de bytes reçues
  AddWriteBytes(0);     //permet de compter le nombre de bytes envoyés
}
```

La propriété "Enabled" permet d'activer ou désactiver un élément

- On va créer deux fonctions pour contrôler l'arrivé ou l'envoi de données
    - Rajouter les éléments suivants dans « private »

```
private:    // Déclarations utilisateur
int FReadCount;
    int FWriteCount;
    void AddReadBytes(int ReadCount);
    void AddWriteBytes(int WriteCount);
    void UpdateComInfo();
```

    - Rajouter les deux fonctions

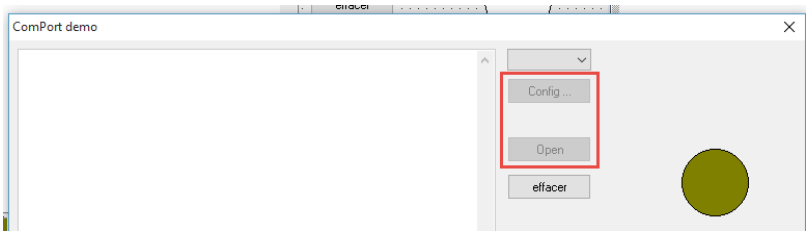
```
 void TForm2::AddReadBytes(int ReadCount)
{
  StatusBar->Panels->Items[0]->Text = "Read bytes: " + IntToStr(FReadCount += ReadCount);
}
//---------------------------------------------------------------------------
void TForm2::AddWriteBytes(int WriteCount)
{
  StatusBar->Panels->Items[1]->Text = "Write bytes: " + IntToStr(FWriteCount += WriteCount);
}
//---------------------------------------------------------------------------
```

La propriété "Enabled" permet d'activer ou désactiver un élément

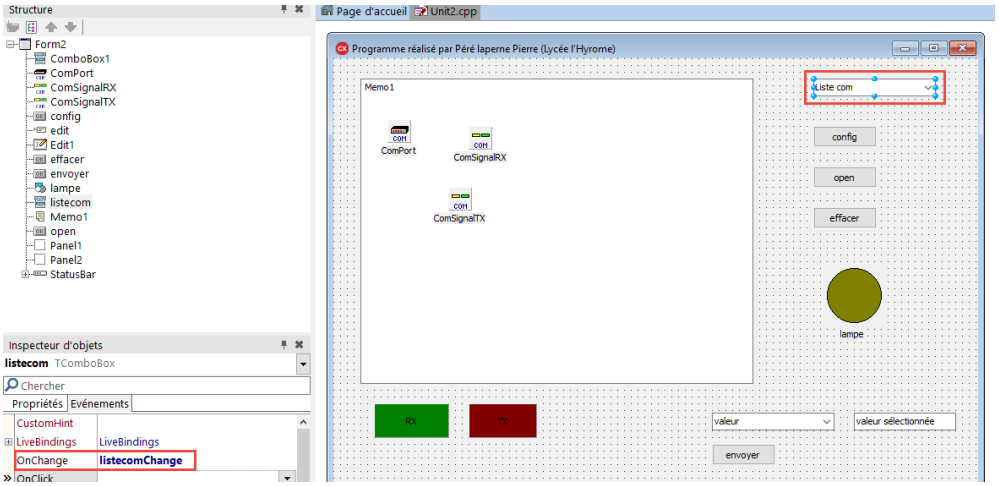
- Brancher la carte Arduino , compiler le programme et contrôler que son numéro de port apparaisse dans la liste

Nom:…… Prénom:……..

- Fermer le programme, débranchez la carte. Ouvrir le programme, les boutons ne doivent plus être accessibles.

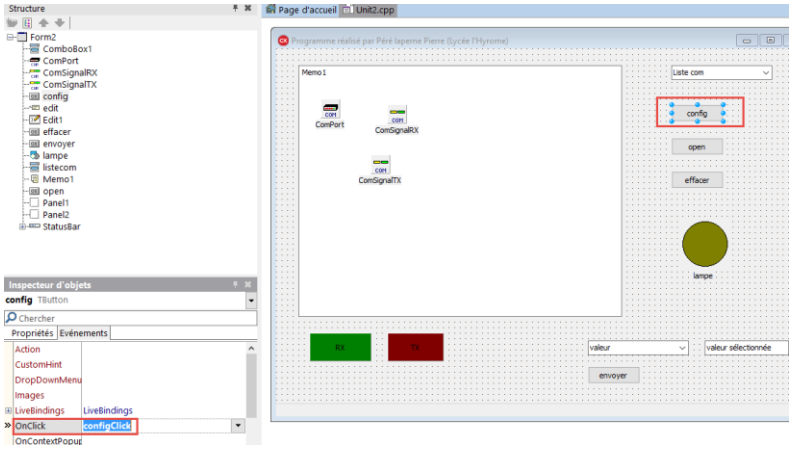


- Rajouter la fonction suivante pour sélectionner le port COM :




```
//
void __fastcall TForm2::listecomChange(TObject *Sender)
{
  ComPort->DeviceName = "\\\\.\\" + listecom->Text;     //enregistre le port COM sélectionné
}
//---------------------------------------------------------------------------
```

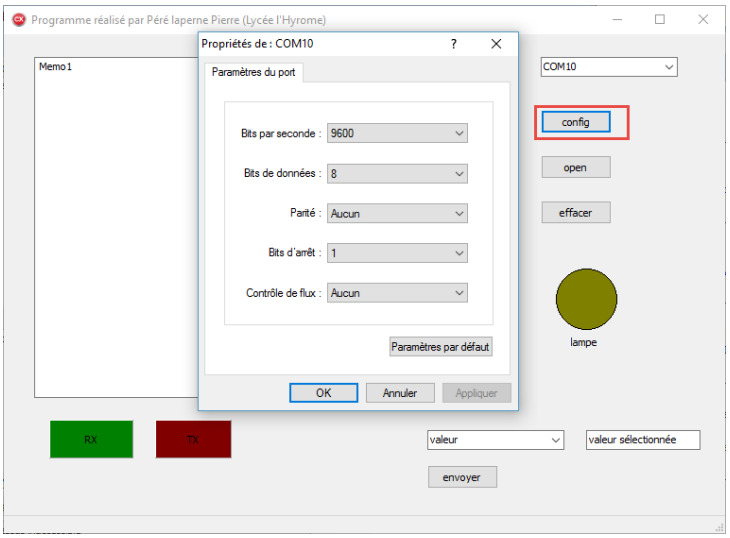
- Rajouter la fonction suivante pour modifier les paramètres du port com sélectionné :



Nom:…… Prénom:……..

```
void __fastcall TForm2::configClick(TObject *Sender)
{
  this->ComPort->ConfigDialog();  //lance la page de configuration windows
}
//---------------------------------------------------------------------------
```

- Brancher la carte Arduino et tester

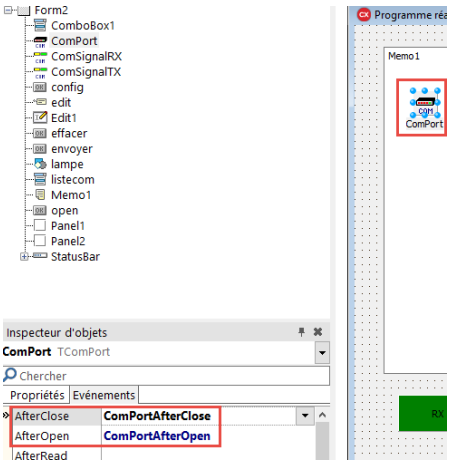


- Modifier le programme de telle manière lorsqu'on sélectionne un chiffre il apparaisse dans l'Edit1 d'à côté.

```
void __fastcall TForm2::ComboBox1Change(TObject *Sender)
{
                :=this->ComboBox1->Text;
}
//---------------------------------------------------------------------------
```

Remarque : on agira dans l'évènement « OnChange »

- On veut modifier le programme à fin que le bouton « envoyer » soit activé lorsqu'on appuie sur le bouton « Open » et se désactive lorsqu'on appuie sur « Close ». Open et Close permettent d'activer ou pas le port.
    - Créer les deux fonctions suivantes

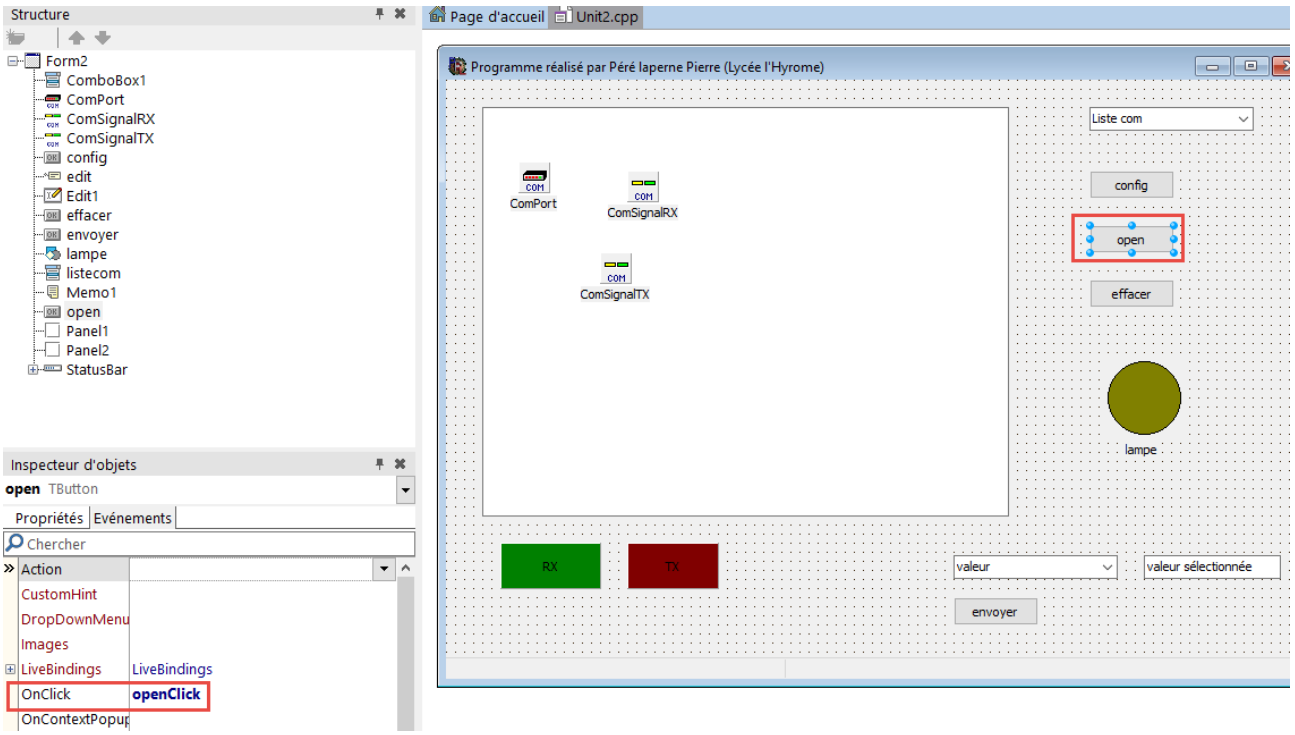


Nom:…… Prénom:……..

- Modifier le programme

```
void __fastcall TForm2::ComPortAfterClose(TCustomComPort *ComPort)
{
 UpdateComInfo();
}
//---------------------------------------------------------------------------

void __fastcall TForm2::ComPortAfterOpen(TCustomComPort *ComPort)
{
  UpdateComInfo();
}
//---------------------------------------------------------------------------
```

- Créer la fonction suivante



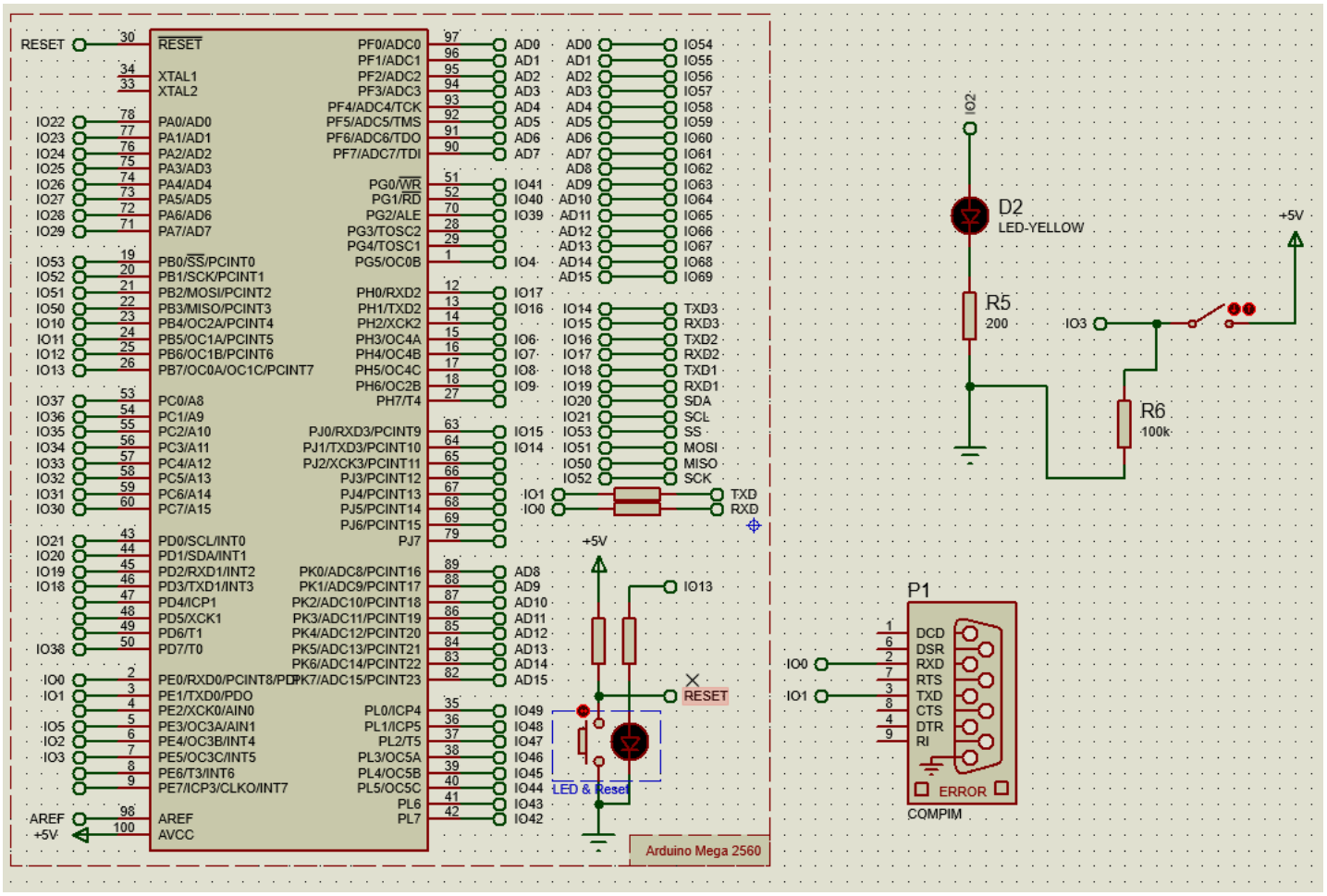
- Modifier le programme

```
void __fastcall TForm2::openClick(TObject *Sender)
{
  ComPort->Active = !ComPort->Active;    //active et désactive le port com sélectionné
}
//---------------------------------------------------------------------------
```

Nom:…… Prénom:……..

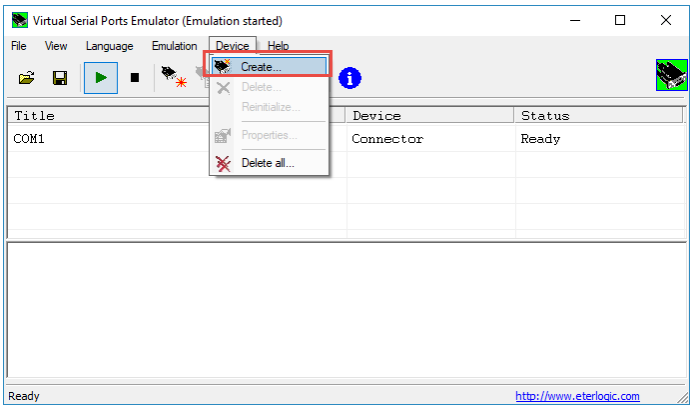
```
void TForm2::UpdateComInfo()
{
  if (Visible)
  {
    if (ComPort->Active)
     this->open->Caption = "Close";
    else
         this->open->Caption = "Open";

    Memo1->Enabled = ComPort->Active;
    if (ComPort->Active)
      Memo1->SetFocus();
    listecom->Enabled = !ComPort->Active;

  }
}
```

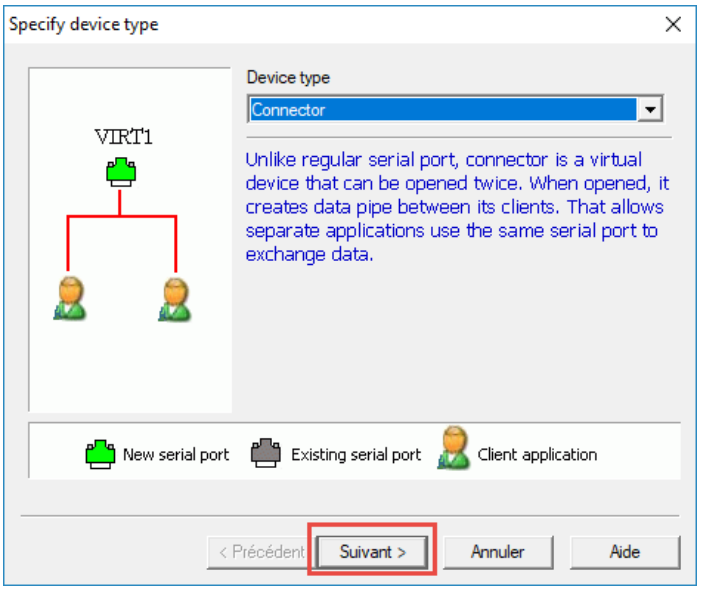
- Réaliser le montage suivant sur Proteus



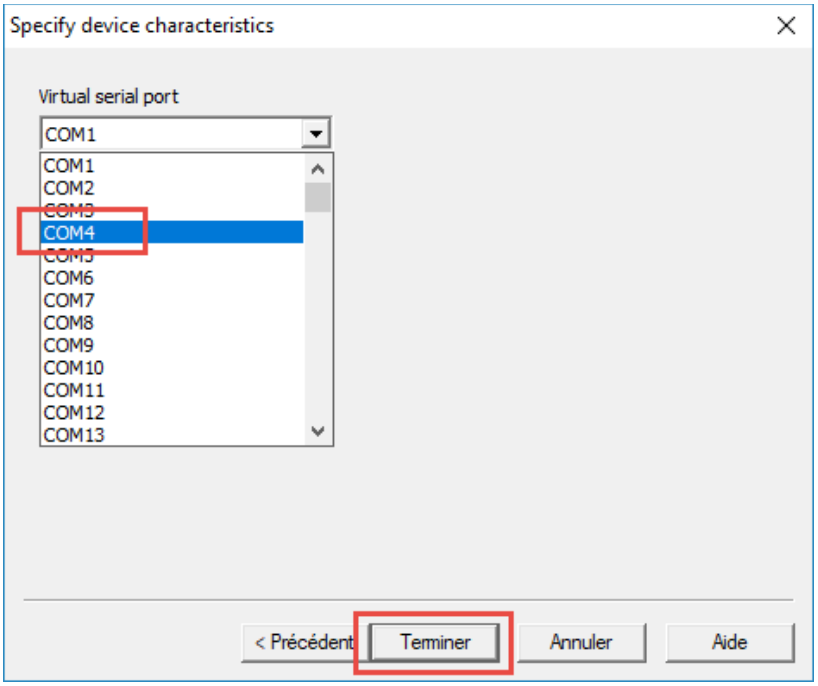
Nom:…… Prénom:……..

- Lancer le logiciel Virtual Serial Port Emulator



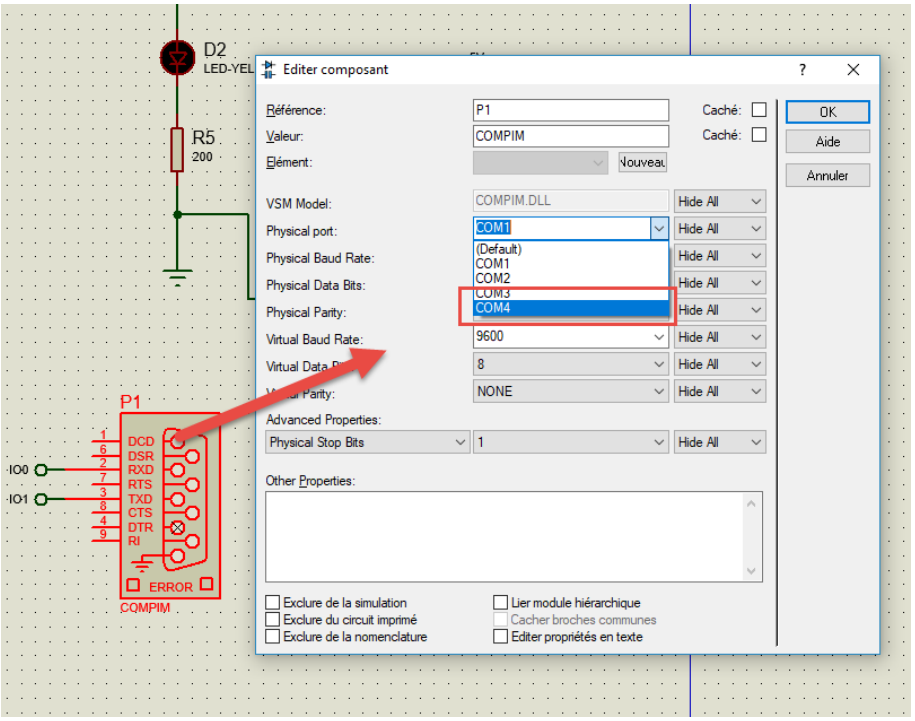


Sélectionner un port



Nom:…… Prénom:……..

Sur Proteus, choisir le Port correspondant :

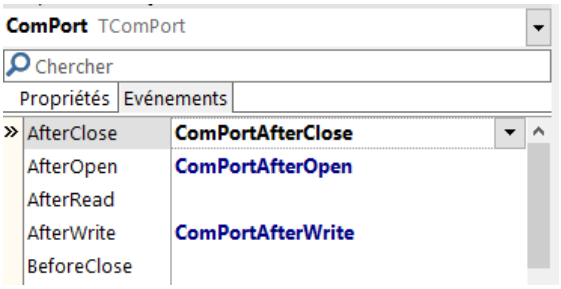


Rentrer le programme suivant sur la carte Arduino :

```
int bouton=0;

int lastbouton=0;

void setup() {

  // put your setup code here, to run once:

Serial.begin(9600);

pinMode(3,INPUT);

pinMode(2,OUTPUT);

}

void loop() {

  // put your main code here, to run repeatedly:

bouton=digitalRead(3);

digitalWrite(2,bouton);

}

}
```

Nom:…… Prénom:……..

- Tester le programme
  - Que se passe t ’il lorsqu’on appuie sur open après avoir sélectionner le port ?
  - Que se passe t ’il lorsqu’on appuie de nouveau sur le même bouton ?
- Modifier le programme à fin qu’on puisse activer, désactiver le bouton envoyer lorsqu’on appuie sur « open » et « close »
- Modifier le programme pour connaître le nombre de bytes envoyés

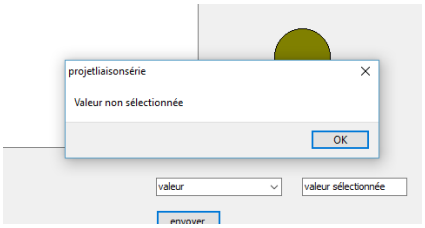



```
void __fastcall TForm2::ComPortAfterWrite(TObject *Sender, Pointer Buffer, int Length,
          bool WaitOnCompletion)
{
  AddWriteBytes(Length);
}
//---------------------------------------------------------------------------
```

- Modifier le programme afin de pouvoir transmettre la valeur sélectionnée à la carte Arduino lorsqu’on appuie sur « envoyer »

```
void __fastcall TForm2::envoyervaleurClick(TObject *Sender)
{
//
if (                                                ') {
ShowMess
}
else
{
this->ComPort->WriteAnsiString(                    )  ;
}
}
//---------------------------------------------------------------------------
```

Attention : Dans le cas où aucune valeur n’est pas sélectionnée, afficher le message « Valeur non sélectionnée »



- Maintenant pour tester le programme, modifier le programme Arduino à fin d’allumer la led quand il reçoit la valeur 1 et l’éteindre quand il reçoit la valeur 0.

<u>Remarque :</u>

Pour lire la valeur reçue, on utilisera la fonction suivante

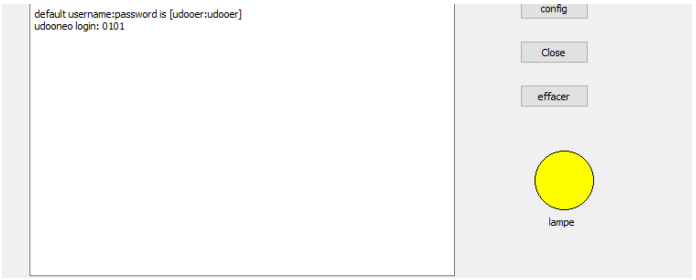
c=Serial.read(); ou c est une variable Char (ce qui permet de récupérer la valeur dans la variable c)

Serial.available() : renvoie la valeur true ou false suivant s’il détecte ou pas une valeur reçue par le port COM

Nom:…… Prénom:……..

- Maintenant on veut récupérer deux valeurs provenant de la carte Arduino pour allumer et éteindre la lampe

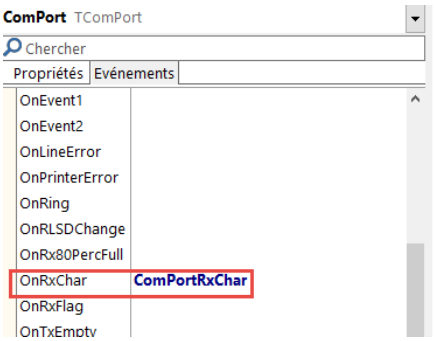
Valeur 1, la lampe s'allume



Valeur 0, la lampe s'éteint

- Modifier le programme C++ Builder pour obtenir ce résultat.

Programme à modifier :




```
void __fastcall TForm2::ComPortRxChar(TObject *Sender)
{
 AnsiString Text = ComPort->ReadAnsiString();   //lit une chaîne de caractère
 valeur=Text;
  Memo1->SelText = Text;
  AddReadBytes(Text.Length());
  if (      [1]==' ') {
                          :=clOlive;
}
if (      [1]==' ') {
                          :=clYellow;
}
}
//---------------------------------------------------------------------------
```

Penser à définir la variable « valeur » dans le programme au format « texte »

Nom:…… Prénom:……..

- Maintenant pour tester le programme, réaliser le programme Arduino à fin d'allumer la lampe quand on appuie sur le bouton1 et l'éteindre lorsqu'on appuie sur le bouton2

  Remarque :

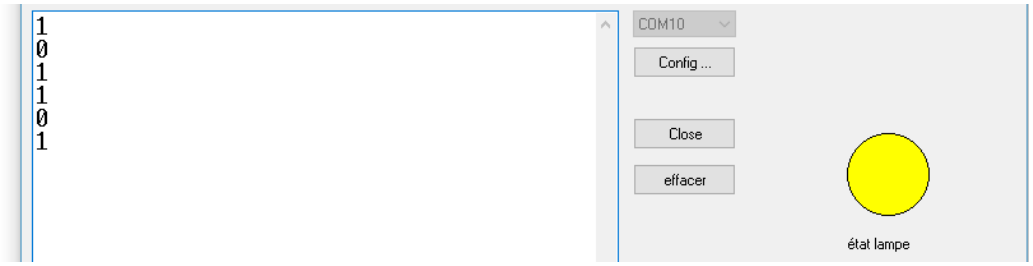
  Pour écrire la valeur qu'on veut envoyer, on utilisera la fonction suivante

  Serial.println(" ");

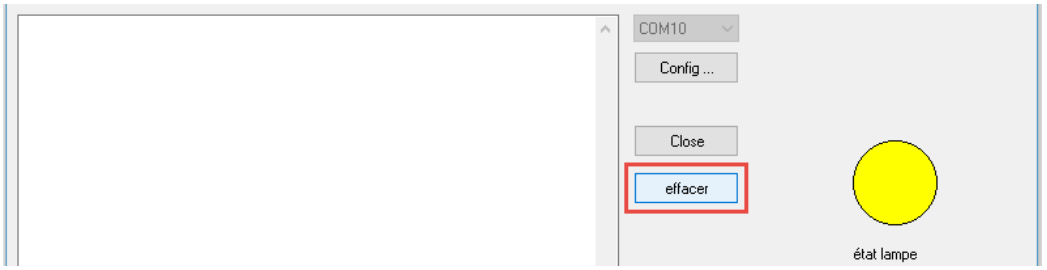
- Pour terminer, modifier le programme C++Builder à fin d'effacer toutes les valeurs sur le Memo lorsqu'on appuie sur le bouton effacer

On utilisera pour cela la propriété Clear(), de Memo

Avant :



Après :



- Vous pouvez récupérer et tester le programme C++ Builder

http://sti2dsinhyrome.fr/doc%20cours/liaison/serie/docserie.html