Nom:…… Prénom:……..

TP SIN
Commander une led depuis Android (Arduino Yun)

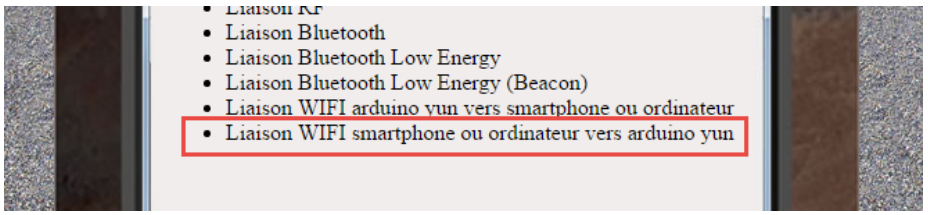
Pré requis (l'élève doit savoir):
Savoir utiliser un ordinateur
Réaliser un programme sur C++ Builder
Réaliser un programme sur Arduino

Programme

Objectif terminal:
L'élève doit être capable de commander un actionneur sur une carte Arduino Yun depuis un smartphone

Matériel

- Ordinateur
- carte Arduino Yun
-deux leds
-servo moteur
-Adresse url du tp sur le site : http://sti2dsinhyrome.fr/docliaison.html



1. **Travail demandé**
   a. Etude du fonctionnement de la carte Arduino

- Programme arduino bridge complet (cf : https://github.com/Anderson69s/Arduino_Yun/tree/master/RGB_LED_Yun_Controller)

```
/*
  Arduino Yun Bridge example
  This example for the Arduino Yun shows how to use the
  Bridge library to access the digital and analog pins
  on the board through REST calls. It demonstrates how
  you can create your own API when using REST style
  calls through the browser.
  Commande possible grâce à ce sketch:
  * "/arduino/digital/13"     -> digitalRead(13)
  * "/arduino/digital/13/1"   -> digitalWrite(13, HIGH)
```

Nom:…… Prénom:……..

```
 * "/arduino/analog/2/123"   -> analogWrite(2, 123)
 * "/arduino/analog/2"       -> analogRead(2)
 * "/arduino/mode/13/input"  -> pinMode(13, INPUT)
 * "/arduino/mode/13/output" -> pinMode(13, OUTPUT)
Cet exemple de code est open-source.
http://arduino.cc/en/Tutorial/Bridge
Modified and translate by Anderson69s
 */


#include <Bridge.h>//On démarre la librarie Bridge pour communiquer entre Linino et atmega32u4
#include <YunServer.h>//On démarre la librarie YunServer qui démarre le serveur coté linino
#include <YunClient.h>//On démarre la librarie YunClient qui démarre le serveur web coté linino


YunServer server;// On écoute sur le port 5555 (par défaut) pour le webserver


void setup() {
 pinMode(13, OUTPUT);//on définit le port 13 en sortie
 digitalWrite(13, LOW);//on éteint la led 13
 Bridge.begin();// On démarre Bridge
 digitalWrite(13, HIGH);//On éclaire la led 13 quand bridge est lancé


 server.listenOnLocalhost(); //On n'accepte que les connexions venant du meme réseau
 server.begin();//On lance le seveur
}


void loop() {
 YunClient client = server.accept();//On accepte les connexions
 if (client) {//Si il y a un nouveau client
  process(client);//On lance la fonction process qui permet de gérer les requêtes du client


  client.stop();//Sinon on ferme la connexion et on libère des ressources
 }


 delay(50); // On fait cela toutes les 50 millisecondes
}


void process(YunClient client) {//Fonction process
 String command = client.readStringUntil('/');//On lit la commande que le client envoie en la découpant à chaque "/"


 if (command == "digital") {//Si c'est une commande digital
  digitalCommand(client);//On lance la fonction digital
 }
```

Nom:…… Prénom:……..

```
  if (command == "analog") {//Si c'est une commande analog
    analogCommand(client);//On lance la fonction analog
  }

  if (command == "mode") {//Si c'est une commande mode pour régler un port en entrée ou sortie
    modeCommand(client);//On lance la fonction mode
  }
}

void digitalCommand(YunClient client) {//Fonction digital
  int pin, value;//on crée deux variables pin et value

  pin = client.parseInt();//On lit la valeur envoyé par le client et on la décompose

  // Si le caractère suivant est un "/", cela veut dire que nous avons une url
  // du type : "/digital/13/1" c'est dans le cas où le port est définit en sortie
  if (client.read() == '/') {
    value = client.parseInt();
    digitalWrite(pin, value);
  }
  else {//Sinon on lit la valeur digital
    value = digitalRead(pin);
  }

  //On envoie des indications au client pour dire que tout va bien
  client.print(F("Pin D"));
  client.print(pin);
  client.print(F(" regle a "));
  client.println(value);

  // On met à jour la valeur entrée dans la mémoire du linino
  String key = "D";
  key += pin;
  Bridge.put(key, String(value));
}

void analogCommand(YunClient client) {//Fonction analog
  int pin, value;//on crée deux variables pin et value

  pin = client.parseInt();//On lit la valeur envoyé par le client et on la décompose
```

```
  // Si le caractère suivant est un "/", cela veut dire que nous avons une url
  // du type : "/digital/9/150" c'est dans le cas où le port est définit en sortie
  if (client.read() == '/') {
    value = client.parseInt();//On lit et on applique la commande
    analogWrite(pin, value);


    //On envoie des indications au client pour dire que tout va bien
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" regle a "));
    client.println(value);


    // On met à jour la valeur entrée dans la mémoire du linino
    String key = "D";
    key += pin;
    Bridge.put(key, String(value));
  }
  else {//Sinon on lit la valeur analog
    value = analogRead(pin);


    //On envoie des indications au client pour dire que tout va bien
    client.print(F("Pin A"));
    client.print(pin);
    client.print(F(" lit un analog de "));
    client.println(value);


    // On met à jour la valeur entrée dans la mémoire du linino
    String key = "A";
    key += pin;
    Bridge.put(key, String(value));
  }
}


void modeCommand(YunClient client) {//Fonction mode
  int pin;//on crée deux variables pin et value
  pin = client.parseInt();//On lit le numéro du pin et on le décompose


  // Si le caractère suinant n'est pas un "/" nous avons une url mal entrée
  if (client.read() != '/') {
    client.println(F("erreur : mauvaise url"));
    return;
  }
```

Nom:…… Prénom:……..

```
  String mode = client.readStringUntil('\r');//On définit une string mode qui lit les valeurs du client


  if (mode == "input") {//Si le mode est input
    pinMode(pin, INPUT);
    //On envoie des indications au client pour dire que tout va bien
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configure en INPUT !"));
    return;
   }


  if (mode == "output") {//Si le mode est output
    pinMode(pin, OUTPUT);
    //On envoie des indications au client pour dire que tout va bien
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configure en OUTPUT !"));
    return;
   }
  //Sinon on envoie un message d'erreur :
   client.print(F("erreur: mode invalide "));
   client.print(mode);
 }
```

- Rentrer ce programme simplifié dans la carte arduino yun

Nom:…… Prénom:……..

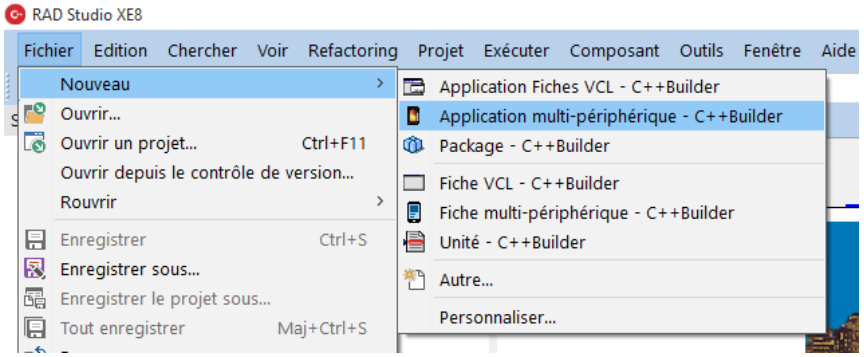
```
#include <Bridge.h>//On démarre la librarie Bridge pour communiquer entre Linino et atmega32u4
#include <YunServer.h>//On démarre la librarie YunServer qui démarre le serveur coté linino
#include <YunClient.h>//On démarre la librarie YunClient qui démarre le serveur web coté linino


YunServer server;// On écoute sur le port 5555 (par défaut) pour le webserver


void setup() {
  pinMode(13, OUTPUT);//on définit le port 13 en sortie
  digitalWrite(13, LOW);//on éteint la led 13
  Bridge.begin();// On démarre Bridge
  digitalWrite(13, HIGH);//On éclaire la led 13 quand bridge est lancé
  delay(2000);
  digitalWrite(13, LOW);//on éteint la led 13

  server.listenOnLocalhost(); //On n'accepte que les connexions venant du même réseau
  server.begin();//On lance le seveur
}


void loop() {
  YunClient client = server.accept();

  if (client) {
    String command = client.readString();
    command.trim();
    Serial.println(command);
    if (command == "ledon") {
      digitalWrite(13,HIGH);
      Serial.println("lampe allumée");
      client.print("lampe allumée");
    }
    else if (command == "ledoff") {
      digitalWrite(13,LOW);
        Serial.println("lampe éteinte");
            client.print("lampe éteinte");
    }
    if (command == "temperature") {
      int val = analogRead(A1);
      client.print(val);
        Serial.println(val);
      }
    client.stop();
  }

  delay(50);
}
```

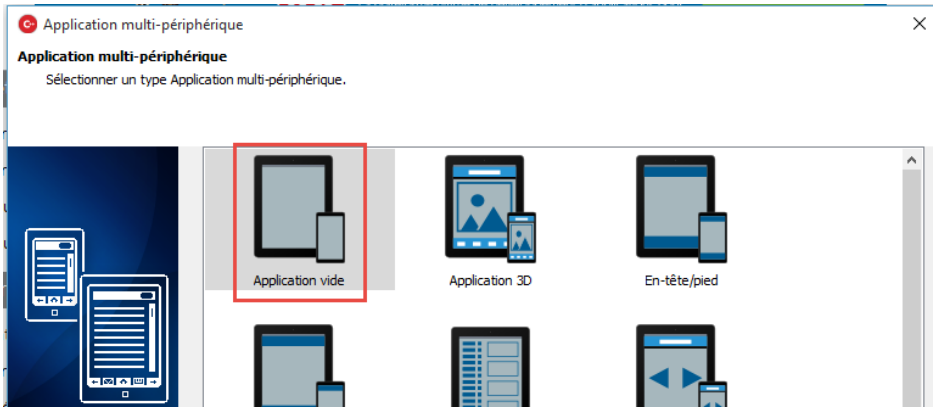
b. Etude de l’application Android

- Lancer C++ Builder

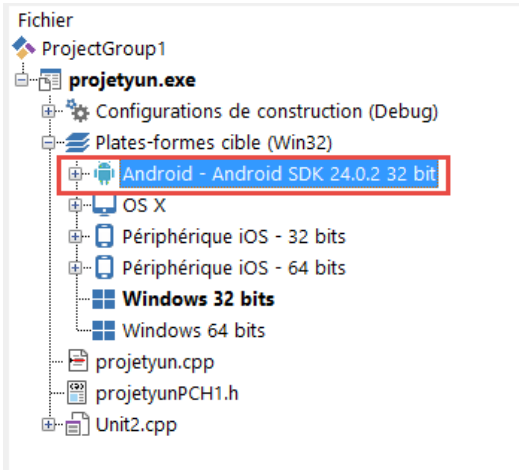
Nom:…… Prénom:……..

- Créer une application Android



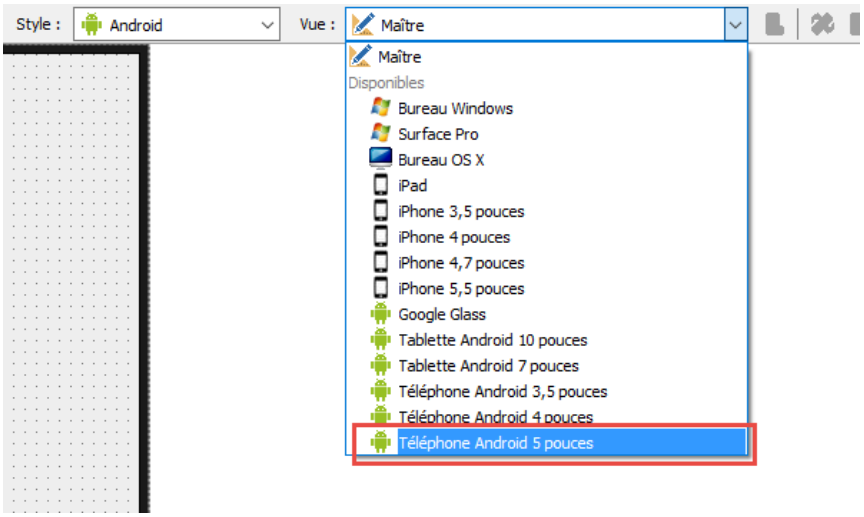


- Sélectionner plateforme cible : Android

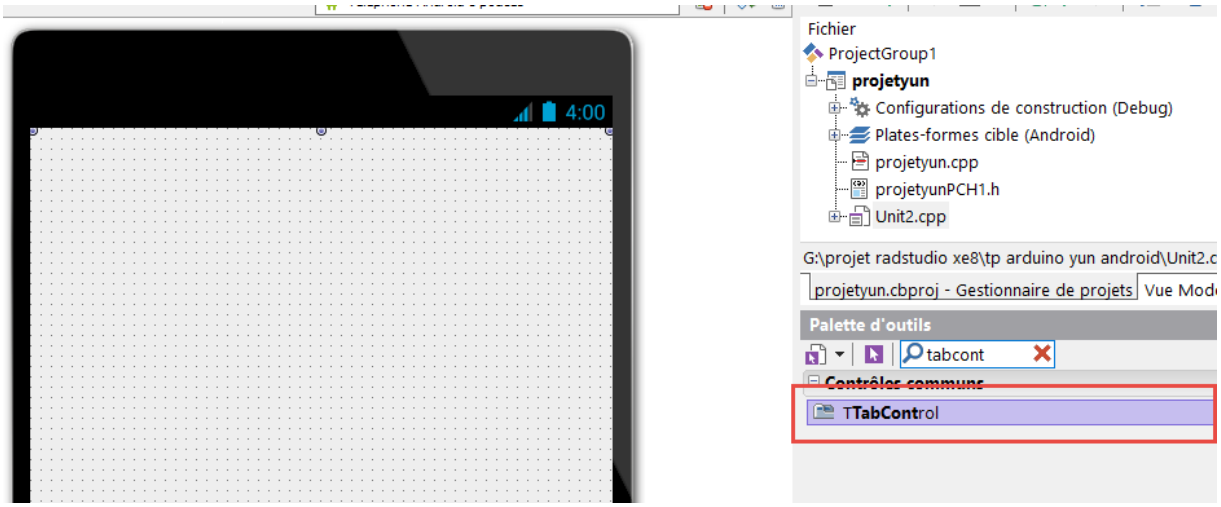


Nom:…… Prénom:……..

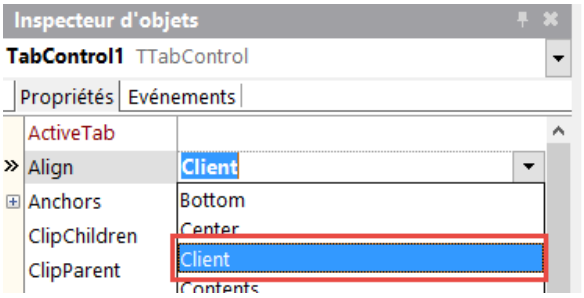
- Puis sélectionner téléphone Android 5 pouces dans Style Android



- Rajouter un TabControl

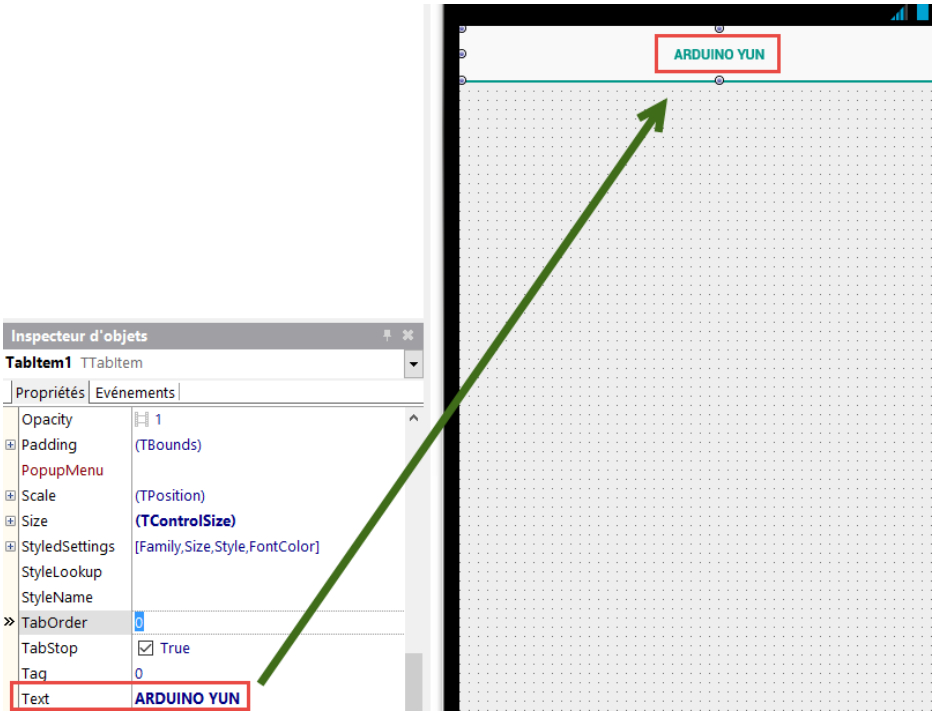


- Modifier la propriété Align

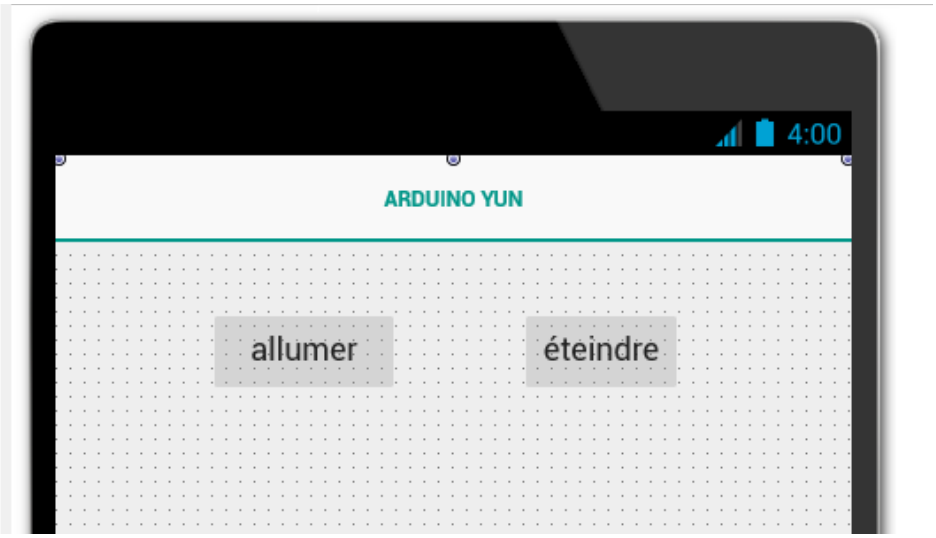


Nom:…… Prénom:……..

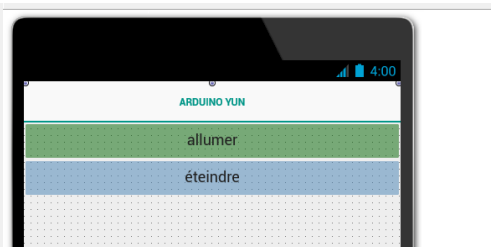
- Modifier le titre



- Rajouter deux boutons

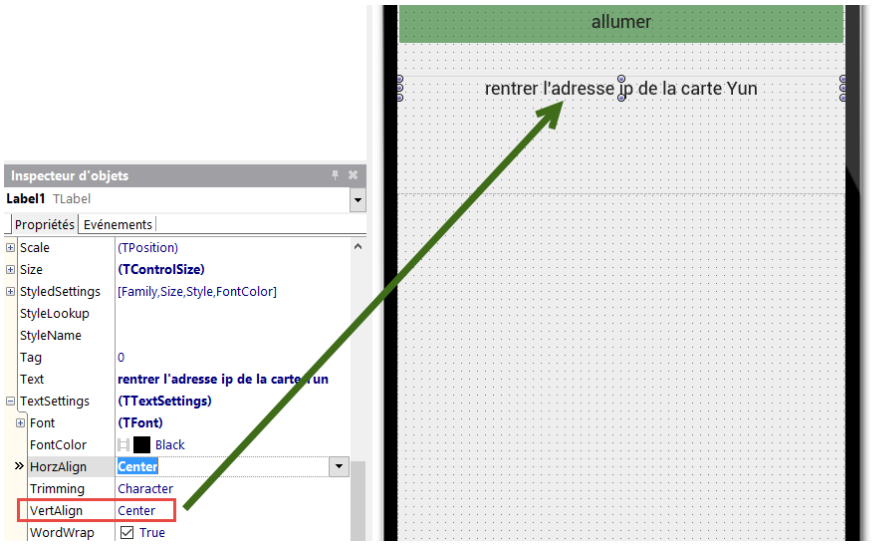


- Modifier le paramètre Align des deux boutons pour qu'ils se trouvent en haut sur toute la largeur du téléphone et modifier les couleurs



Nom:…… Prénom:……..

- Rajouter un label et un edit pour rentrer l'adresse Ip de la carte arduino



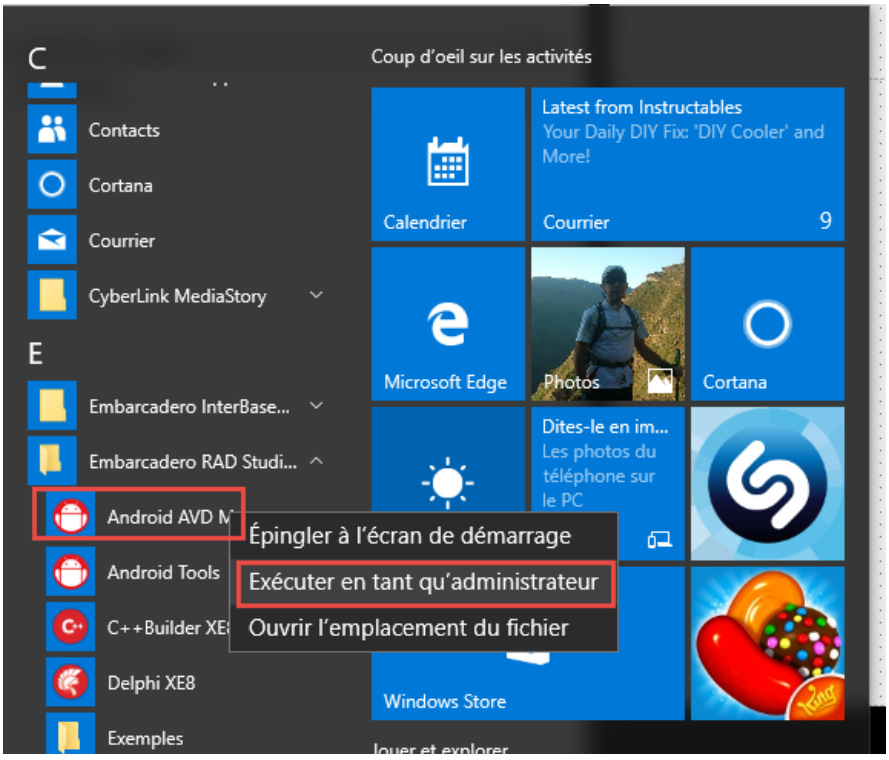
- Rajouter un bouton pour enregistrer l'adresse

Nom:…… Prénom:……..

- Rajouter un webBrowser en dessous du bouton sur le restant du téléphone

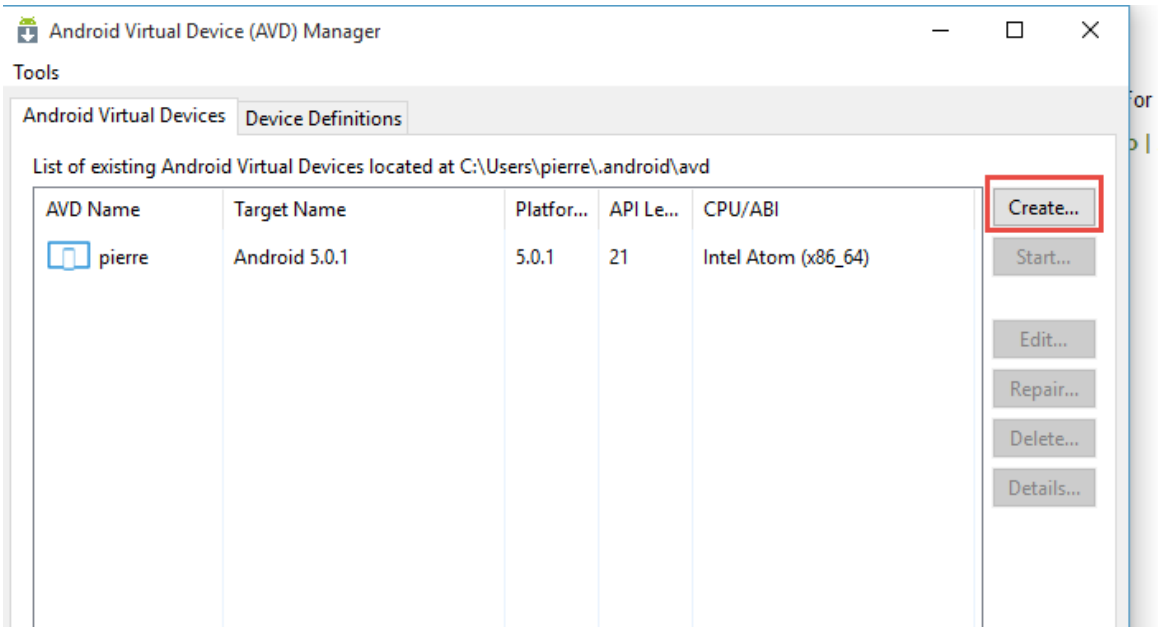
http://docwiki.embarcadero.com/Libraries/XE8/fr/FMX.WebBrowser.TWebBrowser_Methods

- Créer un nouvel émulateur

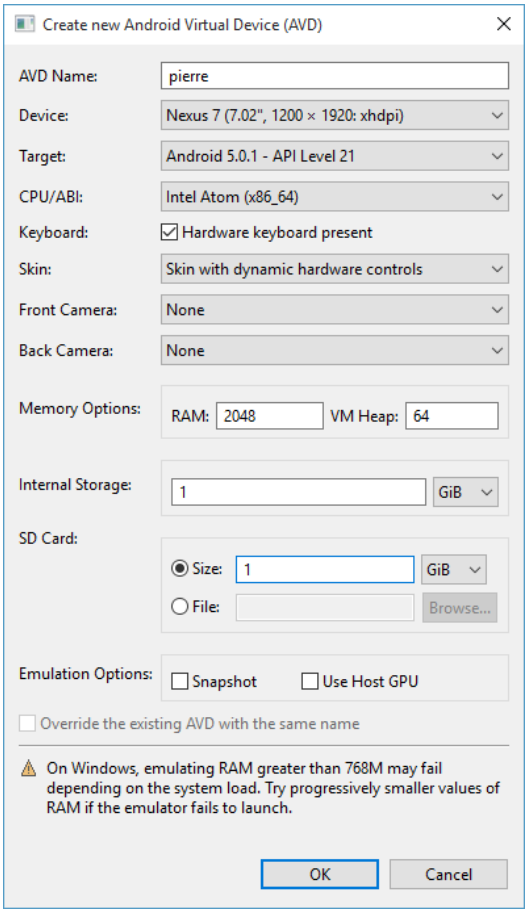


Nom:…… Prénom:……..

- Cliquer sur Create

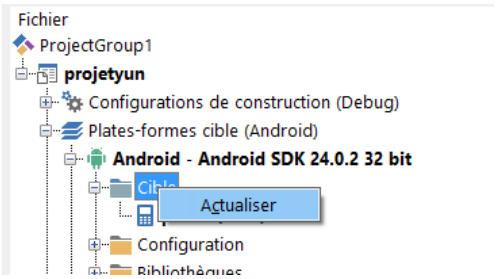


- Configurer votre émulateur



Nom:…… Prénom:……..

- Actualiser la liste (clic droit sur cible))



- Modifier le programme à fin que lorsque la personne a écrit l'adresse IP de la carte arduino yun et qu'elle a rentré une adresse, celle-ci s'enregistre dans une variable « adresseip » lorsqu'on appuie sur le bouton « enregistrer l'adresse ». Puis lancer le navigateur avec l'URL suivante :

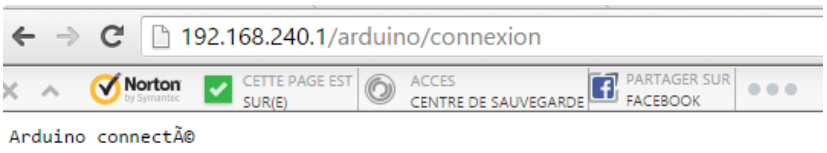
this->WebBrowser1->URL="http://"+adresseip+"/arduino/connexion";

  - De plus si aucune adresse IP n'est rentrée, on doit afficher le message suivant : « Pas d'adresse IP rentrée »

Ligne de programme à remplir :

```
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm2::Button3Click(TObject *Sender)
{

if ((this->Edit1->Text)==  ) {

,
else
{


 this->WebBrowser1->URL="http://"+adresseip+"/arduino/connexion";
 this->WebBrowser1->Navigate();


,
}
//---------------------------------------------------------------------------
```

- Tester le programme. Normalement vous devez voir apparaître sur la page web le message suivant



- Maintenant lorsqu'on appuie sur le bouton « allumer », on veut envoyer l'URL suivante « "http://"+adresseip+"/arduino/ledon" ». Modifier le programme :

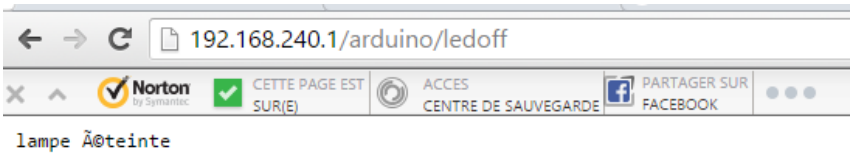
```
void __fastcall TForm2::Button1Click(TObject *Sender)
{



}
```

Nom:…… Prénom:……..

- Pour finir, lorsqu'on appuie sur le bouton « éteindre », on veut envoyer l'URL suivante « "http://"+adresseip+"/arduino/ledoff" ». Modifier le programme :

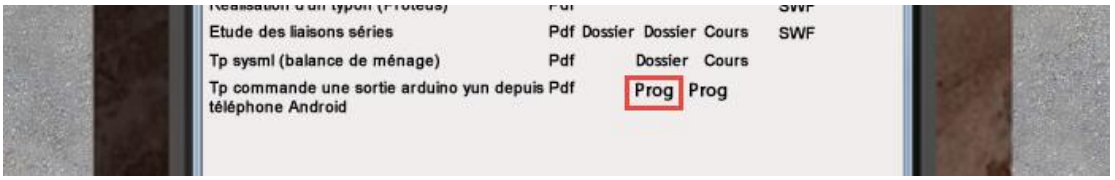
```
void __fastcall TForm2::Button2Click(TObject *Sender)
{



}
//---------------------------------------------------------------------------
```

Remarque :



- Vous pouvez télécharger le programme compilé à l'adresse suivante :

http://www.coursstimartinique.fr/tp%20sin3.html



- Tester plusieurs fois le programme que se passe-t-il ?

- Vider la mémoire du programme et tester le programme de nouveau
- Que faudrait-t 'il rajouter dans le programme pour éviter ce problème

- Rajouter la ligne suivante au bon endroit

this->WebBrowser1->Reload() ;

- A quoi sert cette fonction

Nom:…… Prénom:……..

- Tester de nouveau le programme
- Maintenant on va commander la led depuis des boutons sur la page Web
    - Modifier la page web suivante en rajoutant deux boutons


```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Affichage temperature</title>
 <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
 <script type="text/javascript">
 function refresh() {
 $('#content').load('/arduino/temperature');
 }

</script>
<style type="text/css">
body {
background-image: url(iron02.jpg);

background-repeat: repeat;
}
</style>
</head>
<body onload="setInterval(refresh, 4000);"> /* réactualise la page toutes les 4 secondes
 <p>La valeur de la température est :</p>
<p> <span id="content">Waiting for Arduino...</span></p>
</p>
</body>
</html>
```

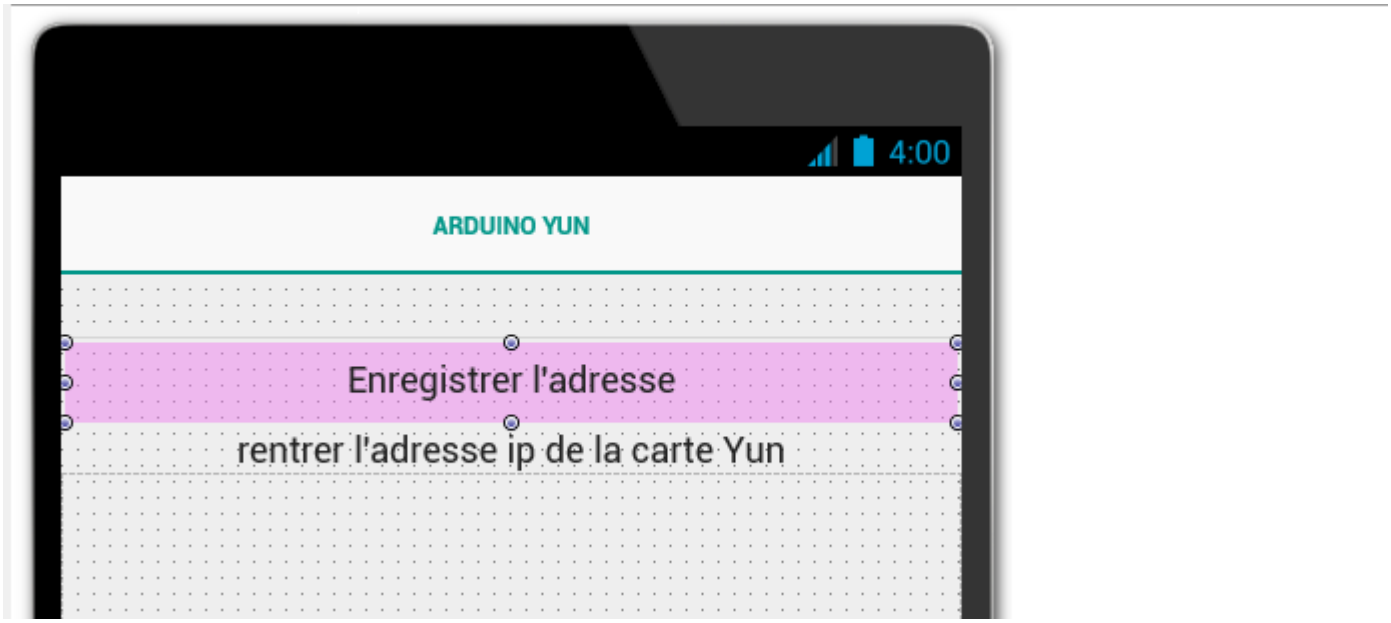
- De plus lorsqu'on appuie sur Led On, on envoie le message « ledon » au serveur. Puis quand on appuie sur Led OFF on envoie le message « ledoff » au serveur.

Remarque :

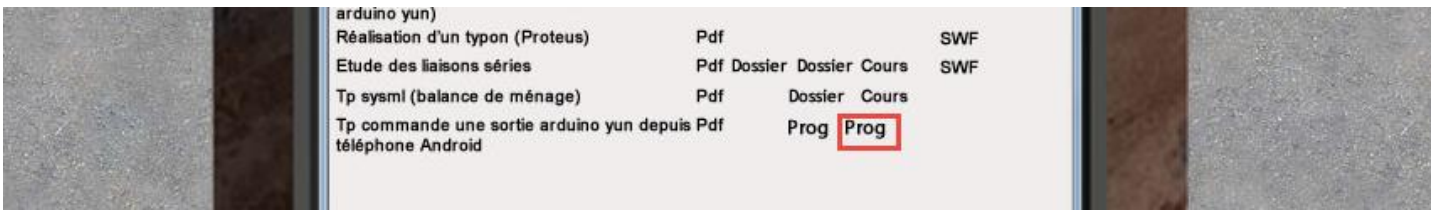
Créer deux fonctions « function ledon() » et « function ledoff() ». De plus rajouter la ligne « Etat de la led depuis carte Arduino Yun:Boutons non actionnés » pour récupérer les messages du serveur.

Nom:…… Prénom:……..

- Faîtes une copie du dossier dans lequel vous avez le programme C++ builder. Renommer le programme « projetyunbis »
- Modifier le programme en rendant invisible les deux boutons



- Modifier l'URL du webBrowser pour se connecter à la page internet du serveur lorsqu'on enregistre l'adresse IP du serveur Yun http://192.168.240.1/sd/indexcommander.php
- Enregistrer la page web sur le serveur et tester l'application
- Pour tester le deuxième programme



Nom:…… Prénom:……..

- Tester le programme plusieurs fois. Que se passe-t-il ?

- Que faudrait-il faire pour éviter ce phénomène ?