

## Application N°4 \_ Commander par un curseur

### Objectif

---

- Commander une lampe à partir d'un Smartphone ANDROID via une carte Arduino. La consigne est donnée par un curseur sur l'écran

### Prérequis

---

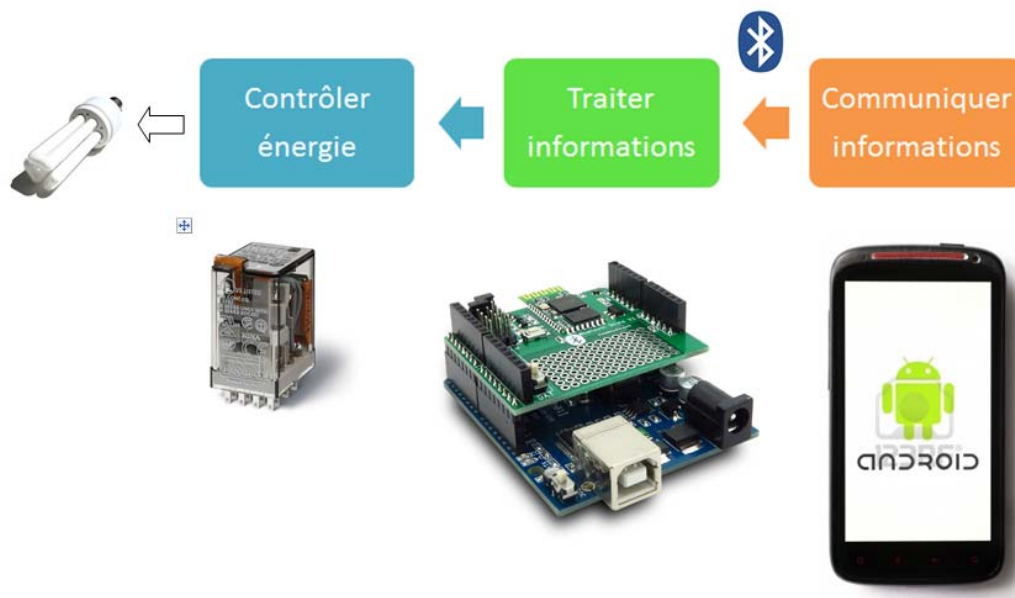
- Configurer la liaison Bluetooth des 2 unités de traitements
  - Voir exemple 1\_AppInv-Arduino\_config Bluetooth

La configuration des liaisons Bluetooth n'est donc pas détaillée dans cette fiche

- Réaliser un programme Arduino traitant les trames de la liaison Bluetooth
  - Voir exemple 2 \_ Commander une lampe

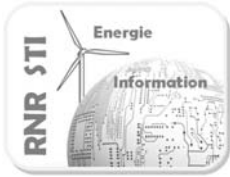
### Configuration

---



Le relais est piloté par la voie 7 du port d'E/S

La led 13 de la carte Arduino sera aussi commandée pour faciliter les essais de mise au point



## Application N°4 \_ Commander par un curseur


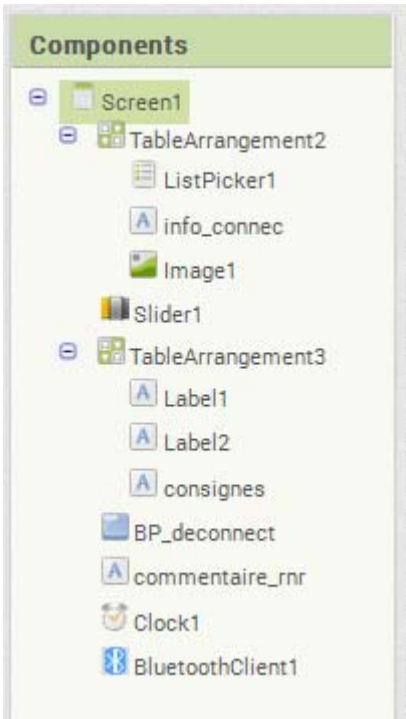
### Programme de la carte Arduino

Programme : <b>_4_arduino_recepteur_consigne_curseur</b>	Commentaires
<pre> //Définition des entrées sorties const int LED_PIN = 13; const int RELAIS=7;  //Déclaration des variables word w; //mot qui reçoit les trames émises  //Initialisation des E/S et communication void setup() { Serial.begin(9600); //vitesse de transmission pinMode(LED_PIN,OUTPUT) ; pinMode(RELAIS,OUTPUT) ; } //Programme principal void loop() {  recevoir();  <b>if (w&gt;120)</b> { digitalWrite(LED_PIN,HIGH); digitalWrite(RELAIS,HIGH); } <b>else</b> { digitalWrite(LED_PIN,LOW); digitalWrite(RELAIS,LOW); } delay(1000); }  //procédure qui lit les trames de la tablette void recevoir(){ if (Serial.available()) { w=Serial.read(); Serial.flush(); } } //The end </pre>	<p>Cette version comporte 2 modifications de la version : _2_arduino_recepteur_on_off</p> <p>« w » va évoluer entre 0 et 255 Configuration du curseur sous APP Inventor</p> <p>La lampe doit être commandée si la valeur est supérieure à 120 (arbitraire).</p> <p><b>Si trame Bluetooth &gt; 120</b></p> <p>Allumer Led Commander le relais</p> <p><b>Sinon</b></p> <p>Eteindre Led Ne plus commander le relais</p>

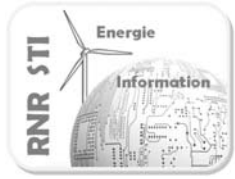


## Application N°4 \_ Commander par un curseur

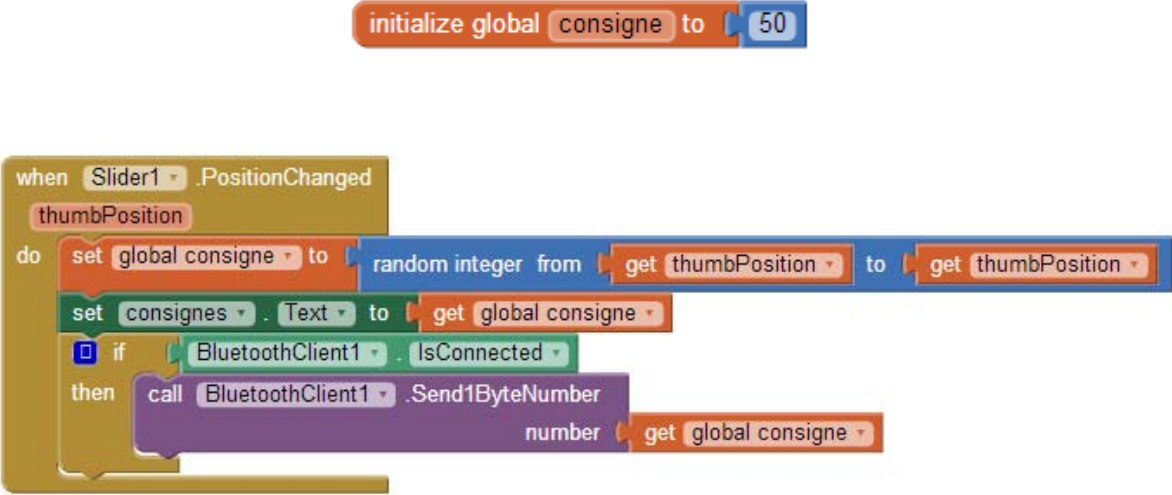
### Application App Inventor du Smartphone

Ecran du terminal	Composants installés
 <p>Commande par potentiometre</p> <p>Se connecter Non connecté</p> <p>valeur de la consigne</p> <p>Se déconnecter RNR STI 2014</p> <p><b>Non-visible components</b> Clock1 BluetoothClient1</p>	 <p>Components</p> <ul style="list-style-type: none"><li>Screen1<ul style="list-style-type: none"><li>TableArrangement2<ul style="list-style-type: none"><li>ListPicker1</li><li>info_connec</li><li>Image1</li><li>Slider1</li></ul></li><li>TableArrangement3<ul style="list-style-type: none"><li>Label1</li><li>Label2</li><li>consignes</li><li>BP_deconnect</li><li>commentaire_rnr</li><li>Clock1</li><li>BluetoothClient1</li></ul></li></ul></li></ul>

Slider est un potentiomètre. Je ne l'ai pas renommé pour faciliter son identification dans les composants.



## Application N°4 \_ Commander par un curseur

Programme de l'application ANDROID	Commentaires
	<p><i>Liaison bluetooth non décrite</i></p> <p>Définition de la variable « consigne » comme un nombre. Prépositionnement à 50 (arbitraire)</p> <p>Lorsque l'évènement «slider1 modifié» apparaît</p> <p>Stocker sa valeur entière dans « consigne »</p> <p>Afficher dans la zone consigne la valeur « consigne »</p> <p>Si connection Bluetooth OK</p> <p>Emettre sur le port Bluetooth la valeur « consigne »</p>

Le programme marche mieux si vous procédez par « touché bref ». En effet si vous faites glisser le curseur, c'est la première acquisition qui est envoyée sur le port Bluetooth. Mon souhait était de faire un programme simple... Piste de solution : cadencement par une seconde horloge pour l'envoi de la consigne