

Application N°7 _ Commande par SMS avec réponse

Objectif

- Commande par SMS d'une lampe via une carte Arduino avec accusé de réception

Un téléphone tiers envoie des SMS « ON » ou « OFF » à votre Smartphone

La carte Arduino renvoie un accusé de l'action par « 1 » ou « 0 »

Le Smartphone transmet cette information par SMS au téléphone émetteur de la commande.

L'état de la lampe est visible sur le Smartphone par un carré jaune ou noir.

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 + exemple 3_ Commande vocale

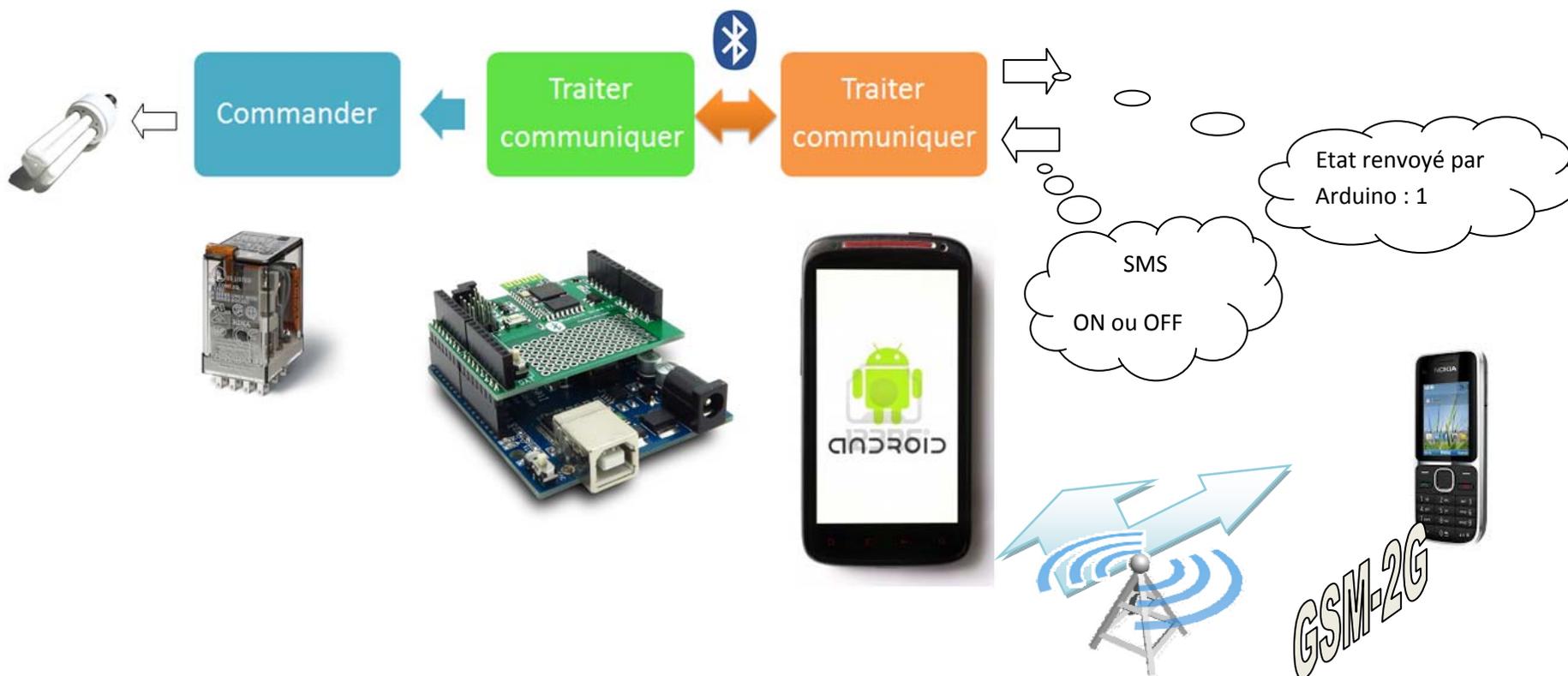
Le programme de la carte Arduino ne sera pas détaillé car identique à l'application N°2

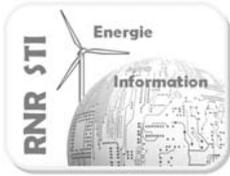
- Réaliser un programme commandant la lampe par SMS
 - Voir exemple 6 _ Commande par SMS



Application N°7 _ Commande par SMS avec réponse

Configuration

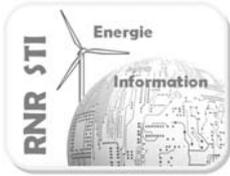




Application N°7 _ Commande par SMS avec réponse

Programme de la carte Arduino

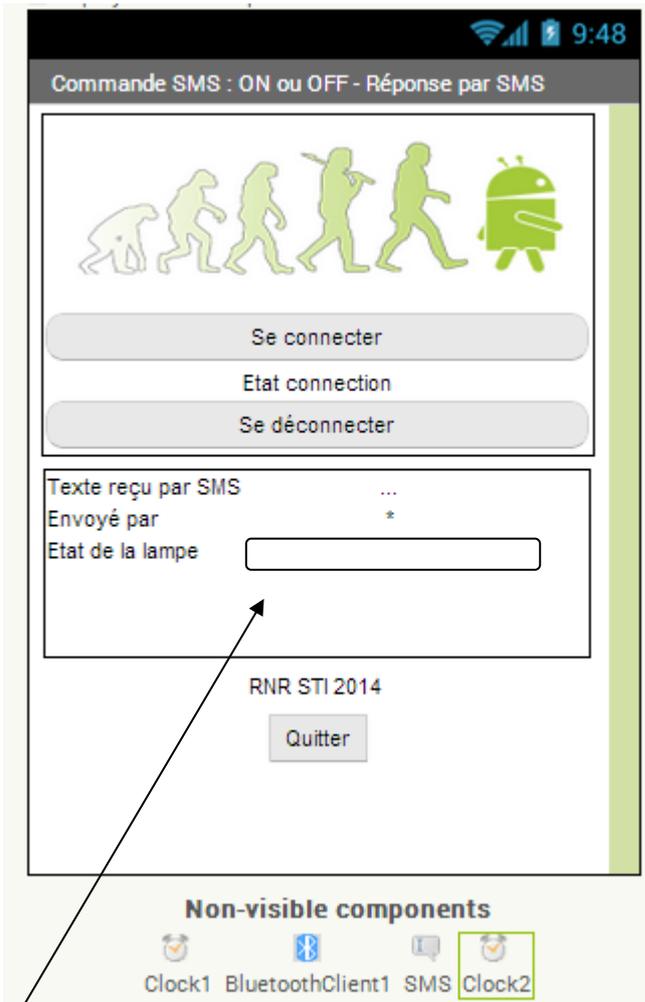
```
//Définition des entrées sorties
const int LED_PIN = 13;
const int RELAIS=7;
//Déclaration des variables
word w; // valeur reçue du smartphone
int reponse; // valeur renvoyée au smartphone
//Initialisation des E/S et communication
void setup() {
  Serial.begin(9600); //vitesse de transmission Bluetooth
  pinMode(LED_PIN,OUTPUT) ;
  pinMode(RELAIS,OUTPUT) ;
}
//Programme principal
void loop() {
  recevoir();
  if (w>120) // vous pouvez aussi mettre ==1 en corrigeant sur les blocs de APP Inventor
  {
    digitalWrite(LED_PIN,HIGH);
    digitalWrite(RELAIS,HIGH);
    reponse=1;
  }
  else
  {
    digitalWrite(LED_PIN,LOW);
    digitalWrite(RELAIS,LOW);
    reponse=0;
  }
  delay(1000);
  envoyer();
}
//procédure qui reçoit de la tablette
void recevoir(){
  if (Serial.available()) {
    w=Serial.read();
    Serial.flush();
  }
}
//procédure qui envoie à la tablette
void envoyer() {
  Serial.print(reponse);
  Serial.flush();
}
//The end
```



Application N°7 _ Commande par SMS avec réponse

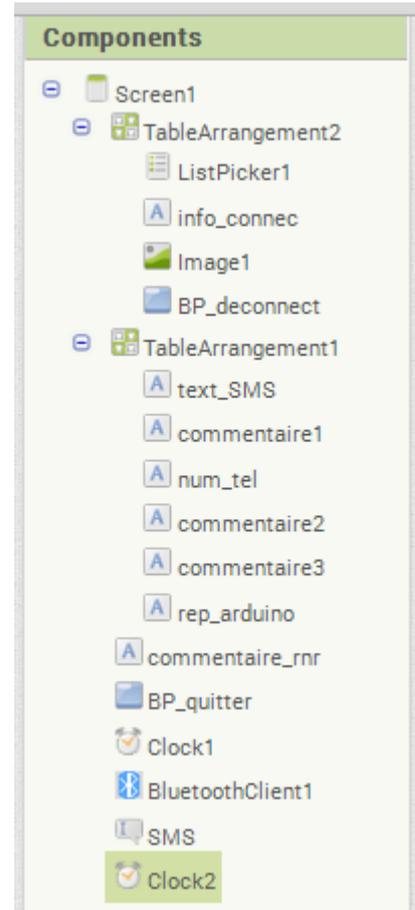
Application App Inventor du Smartphone

Ecran du terminal



L'état de la lampe est visible sur le Smartphone par un carré jaune ou noir.

Composants installés



Properties

Clock2

TimerAlwaysFires



TimerEnabled

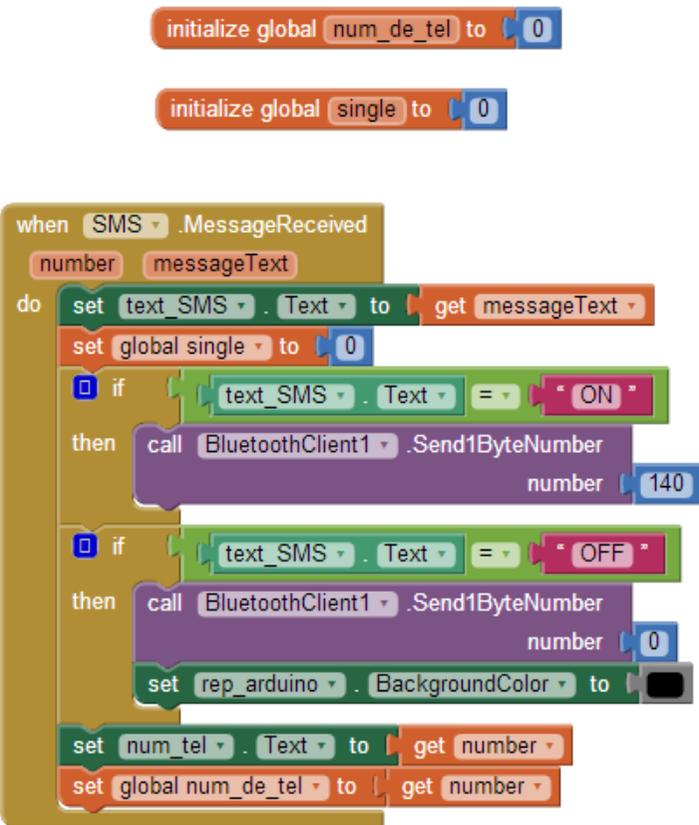


TimerInterval

2000



Application N°7 _ Commande par SMS avec réponse

Programme de l'application ANDROID	Commentaires
 <pre>initialize global num_de_tel to 0 initialize global single to 0 when SMS .MessageReceived number messageText do set text_SMS .Text to get messageText set global single to 0 if text_SMS .Text = "ON" then call BluetoothClient1 .Send1ByteNumber number 140 if text_SMS .Text = "OFF" then call BluetoothClient1 .Send1ByteNumber number 0 set rep_arduino .BackgroundColor to black set num_tel .Text to get number set global num_de_tel to get number</pre>	<p>Définition de la variable « numéro de téléphone » comme un nombre Définition de la variable « single » comme un nombre*</p> <p>Evènement « réception d'un SMS » 2 variables sont alors disponibles : N° tel et contenu du SMS</p> <p>Afficher le contenu du SMS reçu Mettre à zéro « single » // voir commentaire *</p> <p>Si « text_SMS » = «ON » Envoyer la valeur 140 sur le port Bluetooth</p> <p>Si « text_SMS » = «OFF» Envoyer la valeur 0 sur le port Bluetooth</p> <p>L'arrière plan de la zone « rep-arduino » devient noir</p> <p>Afficher le contenu de la variable « num_de_tel » Stocker le N° de l'émetteur dans la variable « num_de_tel »</p>



Application N°7 _ Commande par SMS avec réponse

```
when Clock2.Timer
do
  if text_SMS.Text = "ON"
  then
    if rep_arduino.Text = "1"
    then
      set rep_arduino.BackgroundColor to yellow
      if get global single = 0
      then
        set SMS.PhoneNumber to get global num_de_tel
        set SMS.Message to join ("Etat renvoyé par Arduino : ", rep_arduino.Text)
        call SMS.SendMessage
        set global single to 1
```

La programmation étant événementielle, l'horloge2 a pour fonction de déclencher périodiquement ce traitement

Si « text_SMS » = « ON »

Si « rep_arduino » = « ON » // voir page suivante
L'arrière plan de la zone « rep-arduino » devient jaune

Si single = 0
N° de telephone d'un destinataire de SMS = variable
« num_de_tel »

Text du SMS = « Etat renvoyé ... » + rep_arduino

Envoyer SMS

Single = 1

// Single sera remis à zéro lors de la reception d'un nouveau SMS (page précédente)

*La variable single a pour objet d'éviter d'envoyer une rafale de SMS en retour. Du vécu, au premier essai j'ai reçu 21 SMS d'un coup !!!



Application N°7 _ Commande par SMS avec réponse

```
when Clock1.Timer
do
  if BluetoothClient1.IsConnected
  then
    set info_connec.TextColor to green
    set info_connec.Text to "connecté"
    set rep_arduino.Text to call BluetoothClient1.ReceiveText
    numberOfBytes call BluetoothClient1.BytesAvailableToReceive
  else
    set info_connec.TextColor to red
    set info_connec.Text to "non connecté"
```

Modification de la surveillance de l'état de la connexion

Si la connexion Bluetooth est OK

Lecture des trames émises par Arduino et affichage dans la zone « rep_arduino »