

Projet Arduino

Sommaire :

- Page 3 : Présentation
- Page 4-6 : Montage LED
- Page 7-15 : Montage Récepteur / Emetteur Infra Rouge
- Annexe

Salle Connecté

Le Matériel utilisé :

- Une Carte Arduino Uno
- Une BreadBord
- 18 Jumper Câbles M/M
- Un Module Bluetooth
- Un Emetteur Infra-Rouge
- Un Récepteur Infra-Rouge
- Un Téléphone Android
- Une Led

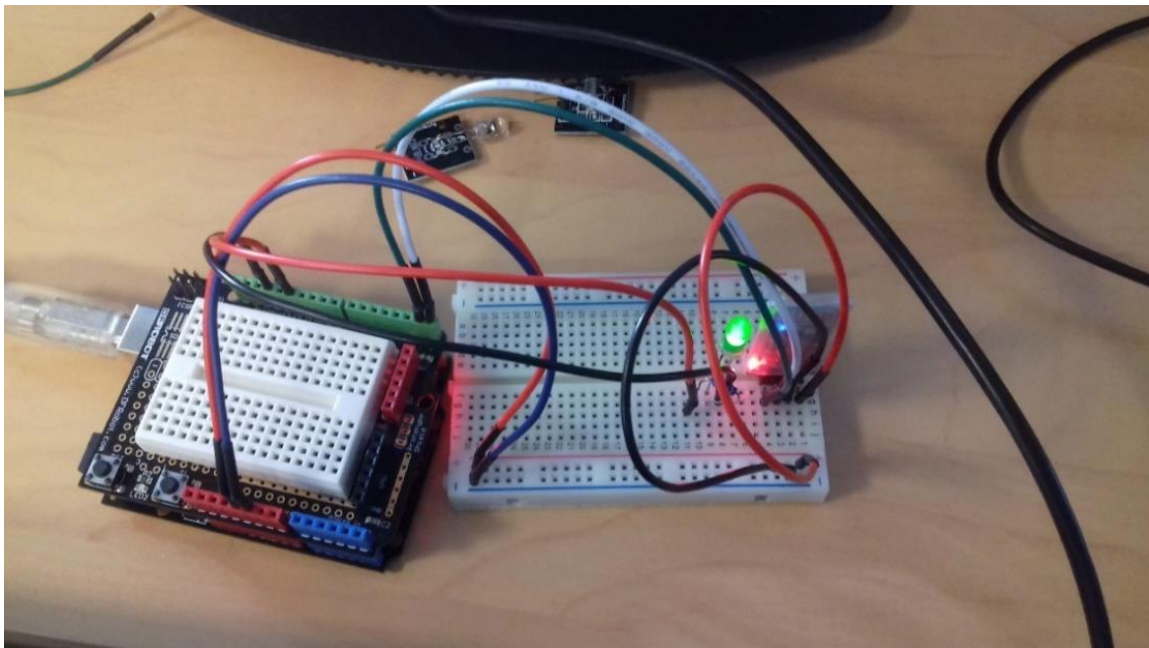
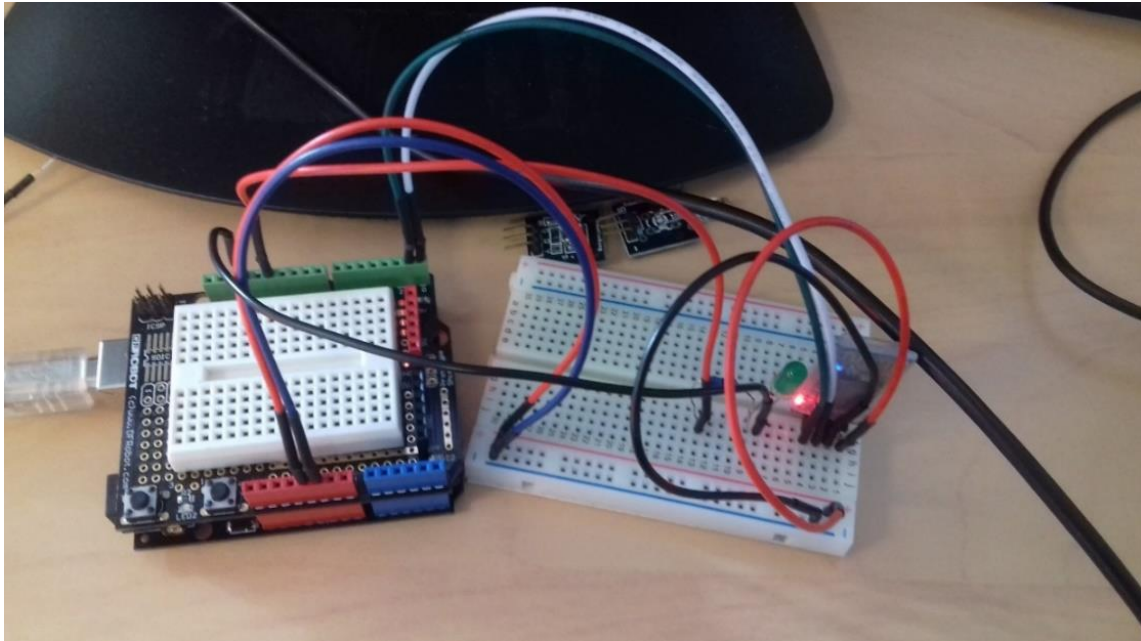
Présentation du projet :

Mon but dans ce projet est de pouvoir contrôler via une application mobile développée a l'aide de **MIT App Inventor 2** une salle avec les appareils suivant :

- Un Lecteur Blu-ray
- 2 Lampes
- Une télévision
- Une banderole LED
- Une chaine Hi-Fi

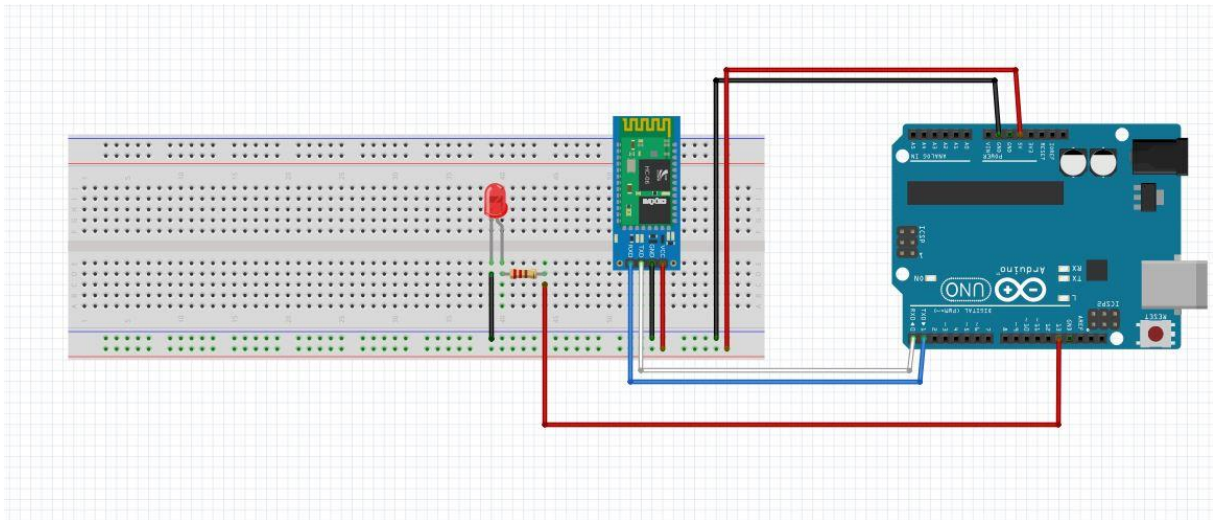
Le téléphone se connecte à l'Arduino en Bluetooth puis l'information est envoyé en Bluetooth et active un signal infra-rouge pour contrôler les appareils.

Montage Led



Le montage Led sert juste à tester si le module Bluetooth fonctionne correctement.

Fritzing



Code Arduino

```
// DEBUT PROGRAMME //

/* -----( Importer la Librairie )-----*/

#include "IRremote.h" // inclue la librairie IRremote.h

/* -----( Declarer les Variables )-----*/

void setup(){ // fonction général

    Serial.begin(9600); // port de communication
    pinMode (13, OUTPUT); // pin de sortie Led

}

void loop() {

    while Serial.available () == 0; // attend de recevoir le caractère
    char c = Serial.read (); // lit les données entrentes
    if (c == 'a') { // caractère pour allumer la Led
        digitalWrite (13, HIGH); // défini la Led en On
    }

    if (c == 'e') { // caractère pour éteindre la Led
        digitalWrite (13, LOW); // défini la Led en Off
    }
}

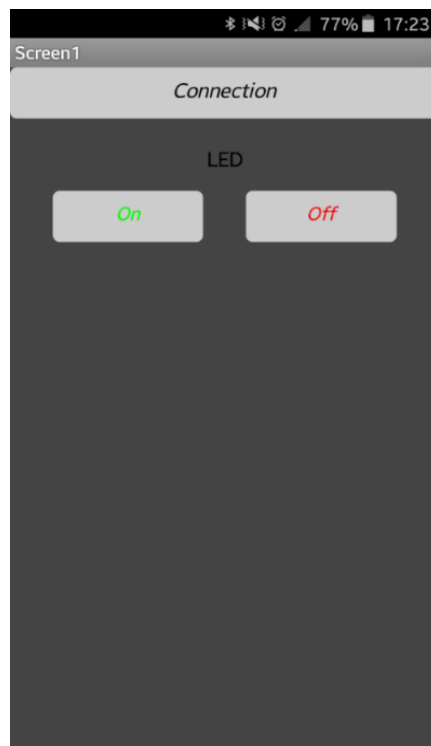
// FIN PROGRAMME //
```

Code App Inventor 2 :

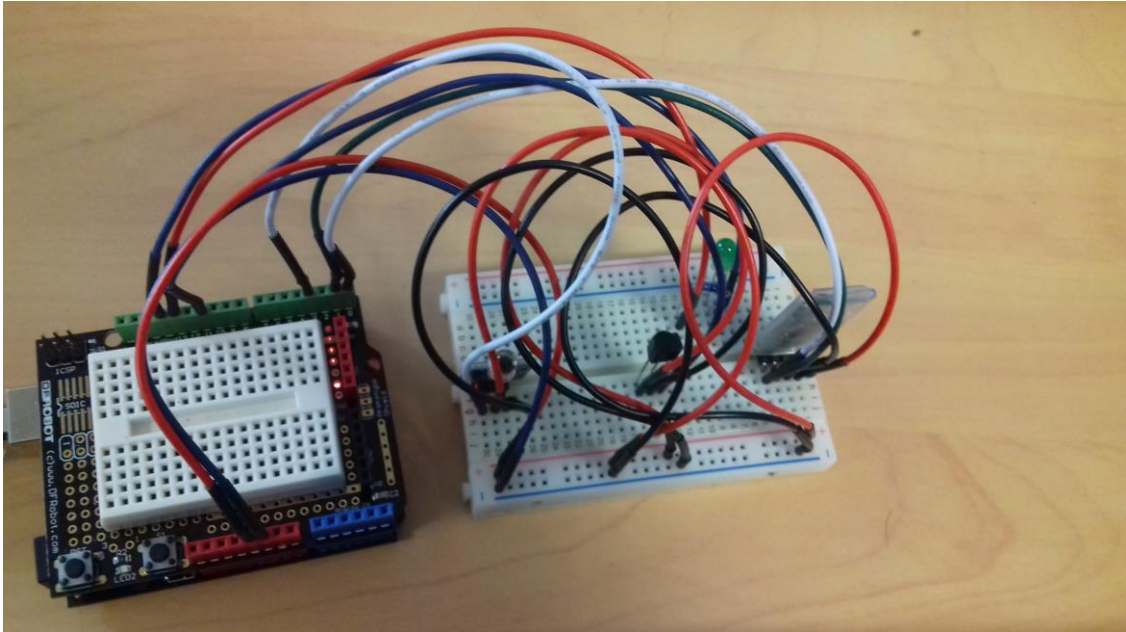
The screenshot displays the App Inventor 2 code editor interface. On the left, the 'Blocs' (Blocks) palette is visible, categorized into 'Incorporé' (Built-in) and 'Screen1'. The 'Incorporé' category includes 'Contrôle' (Control), 'Logique' (Logic), 'Math', 'Texte' (Text), 'Listes' (Lists), 'Couleurs' (Colors), 'Variables', and 'Procédures' (Procedures). The 'Screen1' category includes 'Sélectionneur_de_liste' (List Selector), 'Arrangement_horizont' (Horizontal Arrangement), 'Label1', and 'HorizontalArrangemer' (Horizontal Arrangement). The main workspace shows the 'Interface' section with the following logic blocks:

- quand Sélectionneur_de_liste1 . Avant prise**
 - faire mettre Sélectionneur_de_liste1 . Éléments à Client_Bluetooth1 . Adresses et noms
- quand Sélectionneur_de_liste1 . Après prise**
 - faire mettre Sélectionneur_de_liste1 . Sélection à appeler Client_Bluetooth1 . Se connecter adresse Sélectionneur_de_liste1 . Sélection
 - faire mettre Sélectionneur_de_liste1 . Texte à "Connection"
- quand Bouton1 . Clic**
 - faire appeler Client_Bluetooth1 . Envoyer texte texte "a"
- quand Bouton2 . Clic**
 - faire appeler Client_Bluetooth1 . Envoyer texte texte "e"

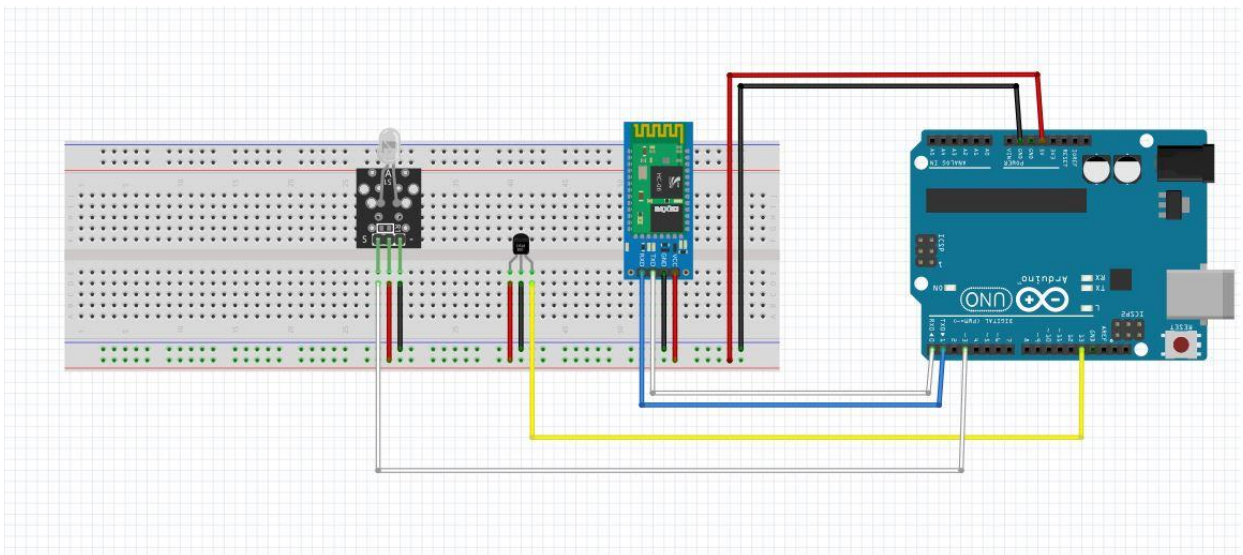
Design App Inventor 2



Détecteur et Emetteur Infra-Rouge :



Fritzing



Code Arduino

```
/*-----( DEBUT PROGRAMME )-----*/

/* -----( Importer la Librairie )-----*/

#include "IRremote.h"                                /* inclue la librairie IRremote.h */

int i = 0;                                           /* initialisation des variables */
char c[3];                                           /* nombre de valeur envoyée pour les appareils ( EX : B11=3 ) */
int receiver = 11;                                   /* pin avec récepteur infra-rouge */

/* -----( Librairie "IRremote.h" )-----*/

IRsend irsend;                                       /* crée un objet pour transmettre des infra-rouge */
IRrecv irrecv(receiver);                             /* crée un objet qui reçoit les informations infra-rouge */
decode_results results;                              /* précise le type de signal infra-rouge */

/* -----( Declarer les Variables )-----*/

void setup(){                                       /* fonction général */

    for(int k = 22; k<30; k++) {pinMode(k, OUTPUT);}

    Serial.begin(9600);                             /* port de communication */
    Serial.println(" || [Bienvenue] || || [Télécommande InfraRouge IR] || "); /* message port COM */
    irrecv.enableIRIn();                             /* écriture des signaux infra-rouge */

    }

/*-----( DEBUT PROTOCOLE IR )-----*/
```

```

void dump(decode_results *results) {                                     /* reconnais les valeurs infra-rouge NEC */

    if (results->decode_type == NEC){                                  /* définis les valeurs reçu en NEC */
        Serial.print("Decoded NEC: ");
    }

    else if (results->decode_type == UNKNOWN) {                       /* définis les valeurs non NEC en inconnue */
        Serial.print("Unknown encoding: ");
    }

}

/*-----( FIN PROTOCOLE IR )-----*/

void loop() {                                                         /* fonction qui se répète */

    if(Serial.available())                                           /* stock les valeurs envoyées */
    {
        c[i] = Serial.read();
        i++;
        Serial.println(c);
    }

}

/*-----( DEBUT IR )-----*/

if (irrecv.decode(&results)) {                                       /* décode les valeurs infra-rouge en hexadeci

```

```
Serial.println(results.value, HEX);
dump(&results);
irrecv.resume();
    }

if(c[0] == 'V') //VIDEOPROJECTEUR (32 bits)
  if(c[1] == '1'){
    if(c[2] == '1') {Serial.println("Touche On Projecteur ");irsend.sendNEC(0x4CB3817E, 32);}
    if(c[2] == '2') {Serial.println("Touche Source Projecteur");irsend.sendNEC(0x4CB3C33C, 32);}
    if(c[2] == '3') {Serial.println("Touche Vga Projecteur");irsend.sendNEC(0x4CB3718E, 32);}
    if(c[2] == '4') {Serial.println("Touche Off Projecteur");irsend.sendNEC(0x4CB3817E, 32);}
    if(c[2] == '5') {Serial.println("Touche Menu Projecteur");irsend.sendNEC(0x4CB311EE, 32);}
  }

if(c[0] == 'L') // LED (32 bits)
  if(c[1] == '1'){
    if(c[2] == '1') {Serial.println("Touche On Led"); irsend.sendNEC(0x40BFA05F, 32);}
    if(c[2] == '2') {Serial.println("Touche Off Led"); irsend.sendNEC(0x40BFB04F, 32);}
    if(c[2] == '3') {Serial.println("Touche Plus Lumière"); irsend.sendNEC(0x40BF50AF, 32);}
    if(c[2] == '4') {Serial.println("Touche Moins Lumière"); irsend.sendNEC(0x40BF10EF, 32);}
  }

if(c[0] == 'T') // TV LG (32 bits)
  if(c[1] == '1'){
    if(c[2] == '1') {Serial.println("Touche Vol+ TV"); irsend.sendNEC(0x20DF40BF, 32);}
    if(c[2] == '2') {Serial.println("Touche Menu TV"); irsend.sendNEC(0x20DFC23D, 32);}
  }

```

```
if(c[2] == '3') {Serial.println("Touche On TV"); irsend.sendNEC(0x20DF10EF, 32);}
if(c[2] == '4') {Serial.println("Touche Vol- TV"); irsend.sendNEC(0x20DFC03F, 32);}
if(c[2] == '5') {Serial.println("Touche 0 TV"); irsend.sendNEC(0x20DF08F7, 32);}
if(c[2] == '6') {Serial.println("Touche 1 TV"); irsend.sendNEC(0x20DF8877, 32);}
if(c[2] == '7') {Serial.println("Touche 2 TV"); irsend.sendNEC(0x20DF48B7, 32);}
if(c[2] == '8') {Serial.println("Touche 3 TV"); irsend.sendNEC(0x20DFC837, 32);}
if(c[2] == '9') {Serial.println("Touche 4 TV"); irsend.sendNEC(0x20DF28D7, 32);}
}
```

```
if(c[1] == '2'){
  if(c[2] == '1') {Serial.println("Touche 5"); irsend.sendNEC(0x20DFA857, 32);}
  if(c[2] == '2') {Serial.println("Touche 6"); irsend.sendNEC(0x20DF6897, 32);}
  if(c[2] == '3') {Serial.println("Touche 7"); irsend.sendNEC(0x20DFE817, 32);}
  if(c[2] == '4') {Serial.println("Touche 8"); irsend.sendNEC(0x20DF18E7, 32);}
  if(c[2] == '5') {Serial.println("Touche 9"); irsend.sendNEC(0x20DF9867, 32);}
}
```

```
if(c[0] == 'B') // BLURAY(32 bits)
  if(c[1] == '1'){
    if(c[2] == '1') {Serial.println("Touche Haut Bluray"); irsend.sendNEC(0xB4B4E21D, 32);}
    if(c[2] == '2') {Serial.println("Touche Bas Bluray"); irsend.sendNEC(0xB4B412ED, 32);}
    if(c[2] == '3') {Serial.println("Touche Ejecter Bluray"); irsend.sendNEC(0xB4B46C93, 32);}
    if(c[2] == '4') {Serial.println("Touche Lecture Bluray"); irsend.sendNEC(0xB4B48C73, 32);}
    if(c[2] == '5') {Serial.println("Touche Pause Bluray"); irsend.sendNEC(0xB4B41CE3, 32);}
    if(c[2] == '6') {Serial.println("Touche On Bluray"); irsend.sendNEC(0xB4B40CF3, 32);}
    if(c[2] == '8') {Serial.println("Touche Droite Bluray"); irsend.sendNEC(0xB4B45AA5, 32);}
    if(c[2] == '9') {Serial.println("Touche Gauche Bluray"); irsend.sendNEC(0xB4B49A65, 32);}
  }
```

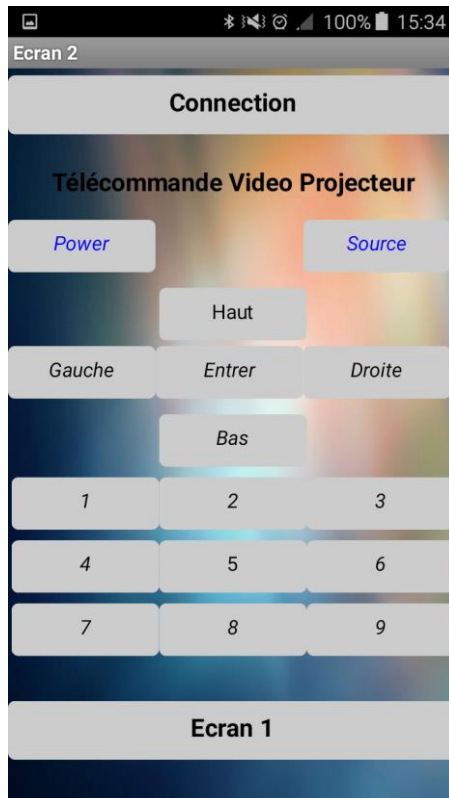
```
if (i == 3) {i = 0; c[0]= '0';c[1]= '0';c[2]= '0';} /* réinitialise nombres à chaque appareil */
```

```
/*-----( FIN IR )-----*/
```

```
}
```

```
/*-----( FIN PROGRAMME )-----*/
```

MIT App Inventor 2



Projet | Screen1 | Add Screen ... | Remove Screen | Designer | Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - Arrangement_horizont
 - Sélectionneur_de_liste
 - Arrangement_horizont
 - Arrangement_horizont
 - Label2
 - Arrangement_horizont
 - Arrangement_horizont

Rename | Delete

Media

ubuntu_1...xrbz.jpg

Viewer

```

when Sélectionneur_de_liste1 BeforePicking
do set Sélectionneur_de_liste1 Elements to Client_Bluetooth1 AddressesAndNames

when Sélectionneur_de_liste1 AfterPicking
do set Sélectionneur_de_liste1 Selection to call Client_Bluetooth1 Connect
address Sélectionneur_de_liste1 Selection

when Bouton21 Click
do open another screen screenName "Screen2"

when Bouton3 Click
do call Client_Bluetooth1 SendText
text "T13"

when Bouton16 Click
do call Client_Bluetooth1 SendText
text "T12"

when Show_Warnings Click
do call Client_Bluetooth1 SendText
text "T11"
  
```

Capture rectangulaire

Activer Windows
Accédez aux paramètres pour activer Windows.

Projet | Screen1 | Add Screen ... | Remove Screen | Designer | Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - Arrangement_horizont
 - Sélectionneur_de_liste
 - Arrangement_horizont
 - Arrangement_horizont
 - Label2
 - Arrangement_horizont
 - Arrangement_horizont

Rename | Delete

Media

ubuntu_1...xrbz.jpg

Viewer

```

when Bouton1 Click
do call Client_Bluetooth1 SendText
text "T15"

when Bouton15 Click
do call Client_Bluetooth1 SendText
text "T14"

when Bouton22 Click
do call Client_Bluetooth1 SendText
text "L11"

when Bouton20 Click
do call Client_Bluetooth1 SendText
text "L12"

when Bouton19 Click
do call Client_Bluetooth1 SendText
text "L13"

when Bouton18 Click
do call Client_Bluetooth1 SendText
text "L14"
  
```

Capture rectangulaire

Activer Windows
Accédez aux paramètres pour activer Windows.

Projet | Screen1 | Add Screen ... | Remove Screen | Designer | Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - Arrangement_horizont
 - Sélectionneur_de_liste
 - Arrangement_horizont
 - Arrangement_horizont
 - Label2
 - Arrangement_horizont
 - Arrangement_horizont

Viewer

```
when Bouton1 .Click
do call Client_Bluetooth1 .SendText
  text " T15 "

when Bouton15 .Click
do call Client_Bluetooth1 .SendText
  text " T14 "

when Bouton22 .Click
do call Client_Bluetooth1 .SendText
  text " L11 "

when Bouton20 .Click
do call Client_Bluetooth1 .SendText
  text " L12 "

when Bouton19 .Click
do call Client_Bluetooth1 .SendText
  text " L13 "

when Bouton18 .Click
do call Client_Bluetooth1 .SendText
  Show Warnings text " L14 "
```

Activer Windows
Accédez aux paramètres pour activer Windows.

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen2
 - HorizontalArranger
 - Sélectionneur_de_liste
 - Arrangement_horizont
 - HorizontalArranger
 - HorizontalArranger
 - Label1
 - Arrangement_horizont
 - HorizontalArranger

Viewer

```
when Bouton18 .Click
do call Client_Bluetooth1 .SendText
  text " B16 "

when Bouton1 .Click
do call Client_Bluetooth1 .SendText
  text " B14 "

when Bouton19 .Click
do call Client_Bluetooth1 .SendText
  text " B15 "

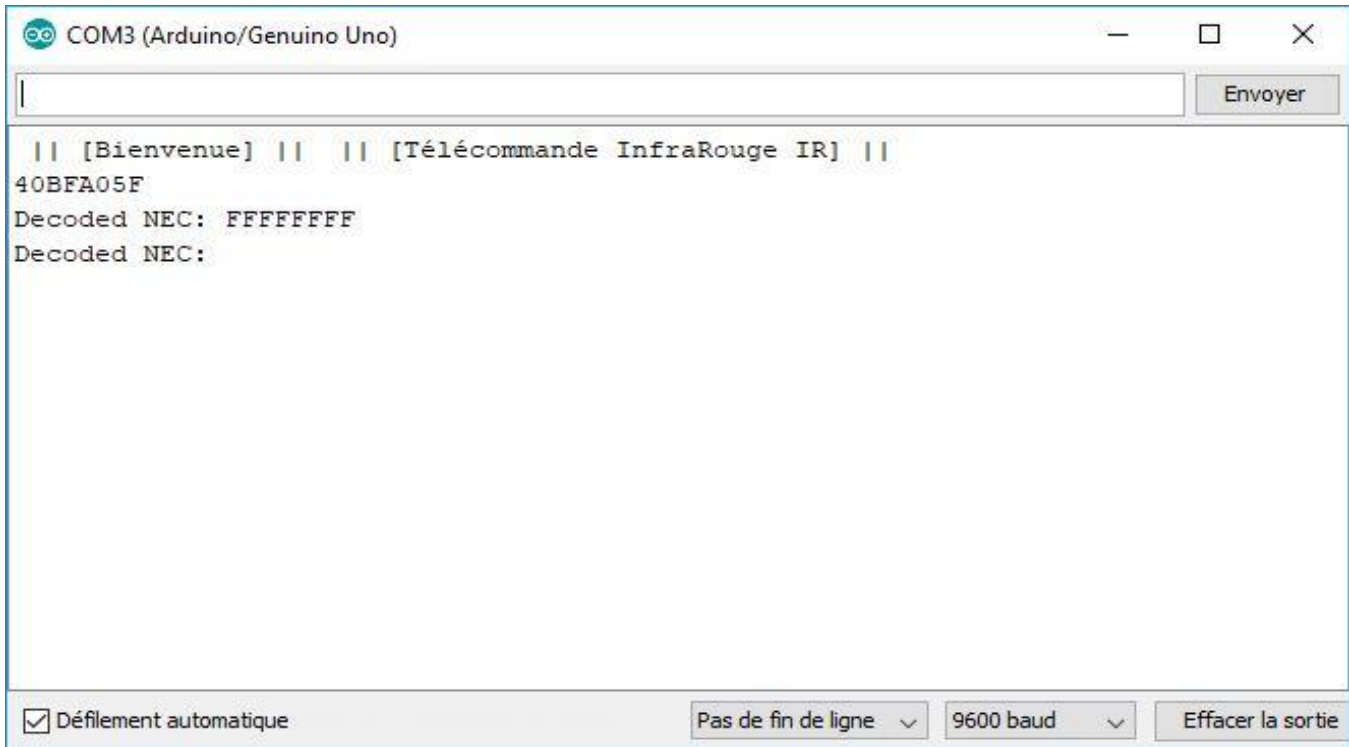
when Bouton21 .Click
do call Client_Bluetooth1 .SendText
  text " B11 "

when Bouton20 .Click
do call Client_Bluetooth1 .SendText
  text " B13 "

when Bouton24 .Click
do call Client_Bluetooth1 .SendText
  Show Warnings text " B18 "
```

Activer Windows
Accédez aux paramètres pour activer Windows.

Port COM



Chaque touche de l'application est synchronisée avec une valeur infra-rouge.

EX : NEC 0x40BFA05F :

- NEC = Protocole infra-rouge.
- 0x40BFA05F = valeur IR touche On de la télécommande banderole LED.

Annexe :

<https://openclassrooms.com/courses/perfectionnez-vous-dans-la-programmation-arduino/codez-et-decodez-des-informations>

<https://letmeknow.fr/blog/2013/07/01/tuto-ajouter-le-bluetooth-au-arduino/>

<https://www.kzenjoy.net/2015/votre-telecommande-universelle-grace-a-arduino/>

<http://www.instructables.com/id/Remotely-Control-LED-using-HC-05-Bluetooth-Arduino/>