

Guirlande commandée par Bluetooth

C'est une bande de leds commandée par Bluetooth avec une application Android, pour créer une guirlande pilotable. ATTENTION !!! Le prix affiché correspond à l'achat de tout le matériel électronique, bien sûr l'alimentation est de la récupération, donc est gratuite.

 Difficulté Facile

 Durée 2 jour(s)

 Catégories Décoration, Électronique, Robotique

 Coût 90 EUR (€)

Sommaire

Introduction

Étape 1 - Le code arduino

Étape 2 - Créer son application Android

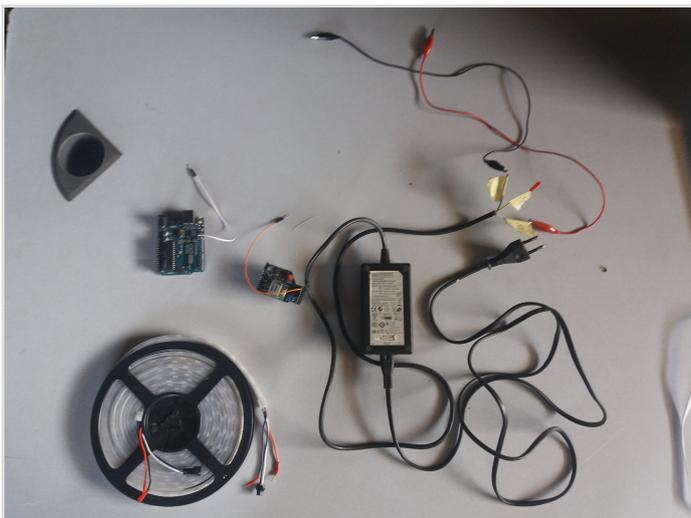
Étape 3 - La mise en route

Notes et références

Commentaires

Introduction

Ce projet permet de réaliser une guirlande avec une bande de leds. Cette guirlande est reliée à une carte Arduino uno, qui est connectée à un module Bluetooth, pouvant être piloté à distance par un téléphone portable. Dans ce projet vous allez créer une application Android, apprendre à utiliser une bande de leds...



Matériaux

Une bande de Led, ici de la marque Adafruit néopixel ;
Une carte arduino uno ;
Un module Bluetooth modulable avec une carte arduino uno (ici un module FB155BC) ;
Une alimentation fournissant du 5V (ici une alimentation d'ordinateur modifiée pour le projet) ;

Outils

Un ordinateur ;
Un téléphone portable ;



<http://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/techniques/3396/3396-1-appinv-arduino-config-bluetooth.pdf>

 code du bluetooth pour la carte arduino

 Code pour le contrôle de leds

 Code pour la carte arduino

Étape 1 - Le code arduino

Dans cette étape vous allez programmer la carte Arduino uno. Le mieux est de tester chaque code pour comprendre son fonctionnement, il faut logiquement étudier le code rassemblant le Bluetooth et l'activation de la bande de leds en derniers.

```
simple: Arduino 1.8.5
Fisher: Editor: Create: Quick: Help

simple

// Modified from simple sketch (c) 2012 Dan Erusso
// Released under the GNU GPL license to match the rest of the Adafruit_MeowFart library

#include <Adafruit_MeowFart.h>
#define _PIN_
#include <avr/power.h>
#include <math.h>

// Which pin on the Arduino is connected to the MeowFart?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN 0

// How many MeowFarts are attached to the Arduino?
#define NUMMEOWFARTS 16

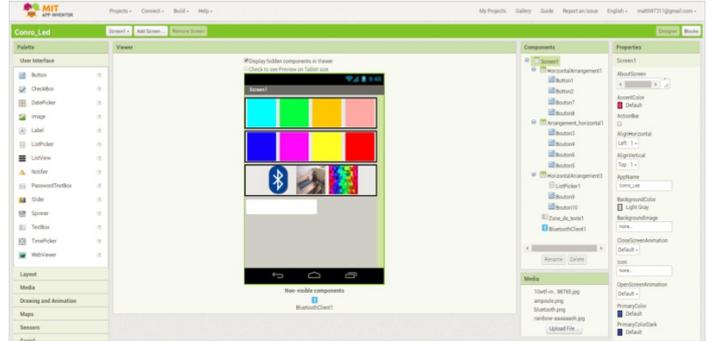
// When we setup the MeowFart library, we tell it how many pins, and which pin to use to send signals.
// Note that for older MeowFart strips you might need to change the third parameter--see the strandtest
// example for more information on possible values.
Adafruit_MeowFart meowfart = Adafruit_MeowFart(NUMMEOWFARTS, PIN, MEOW_ON = MEOW_ON2010);

int delayval = 500; // delay for half a second

void setup() {
  // Note that for older Trinkets or Gemmas, you can remove these three lines if you are not using a Trinket
  #if defined (_PIN_ || ATmega168)
  if (!digitalWrite(PIN, LOW)) pinMode(PIN, OUTPUT);
  #endif
  #define MEOWFARTS
  #endif
  // End of trinket special code
}
```

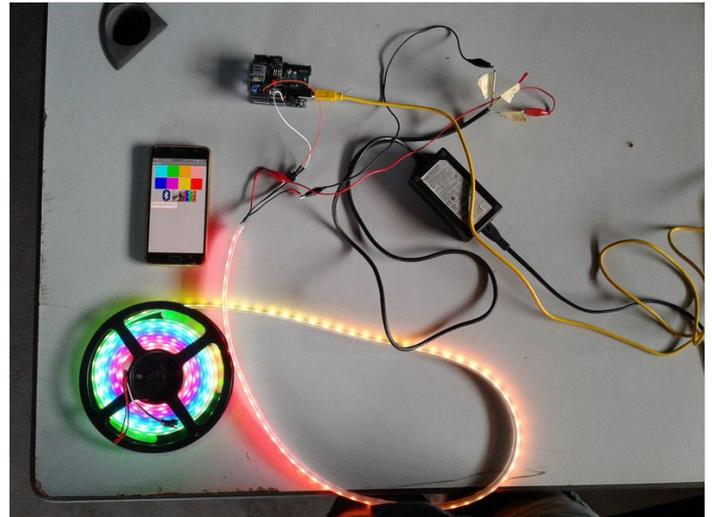
Étape 2 - Créer son application Android

Pour créer votre application il vous faut aller sur MIT App Inventor 2, le logiciel est très facile de prise en main. C'est à vous de jouer avec le logiciel pour comprendre comment il marche, une petite aide quand même, allé sur le lien "eduscol" pour un rapide cours sur l'utilisation du Bluetooth sur App Inventor.



Étape 3 - La mise en route

La mise en route est rapide, il vous faut tester la communication entre votre tel et la carte Arduino. pour cela il est préférable de faire afficher un message dans le moniteur série de l'interface Arduino. Une fois la communication assimilée il vous faut juste envoyer des chaînes de caractères, qui sont lus par la carte Arduino et en fonction de ces caractères enverra une couleur à la bande.



Notes et références

La contrainte la plus importante est de trouver une alimentation pour la bande de leds, car la carte arduino ne supporte pas la bande de leds entièrement allumée.

Le téléphone utilisé est le mien donc il ne m'a rien coûté.