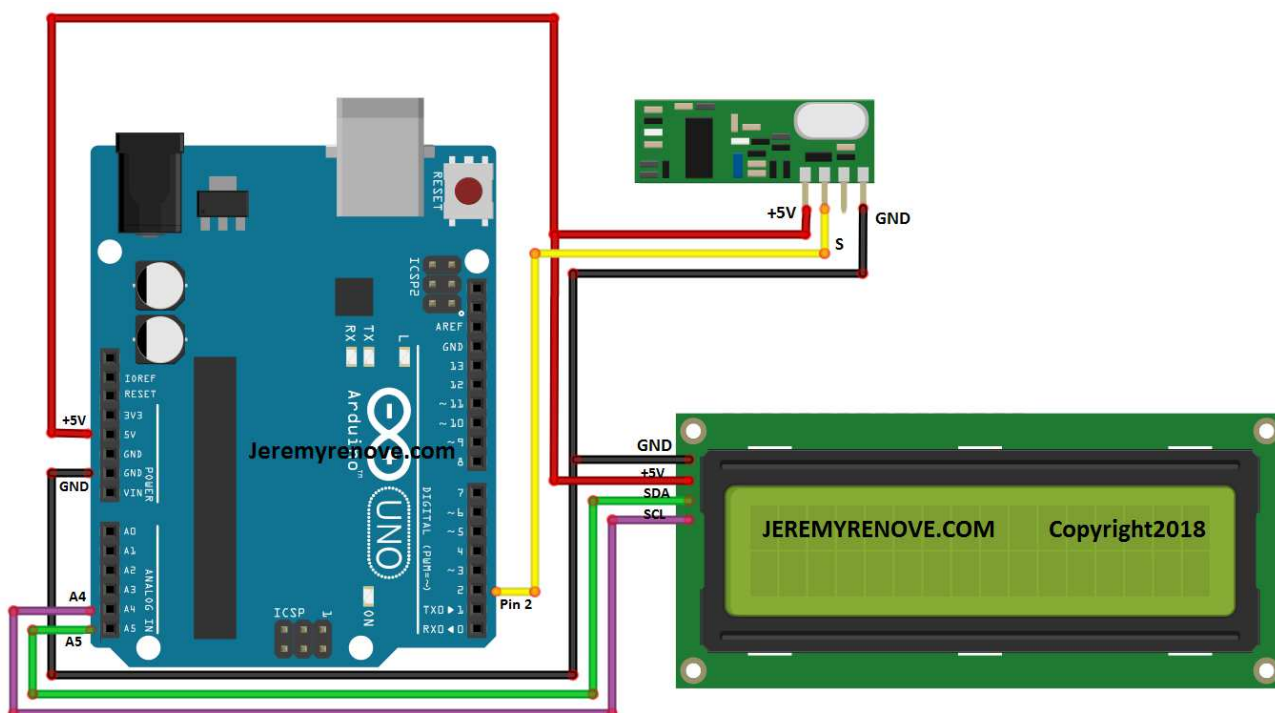


Bonus Jeremyrenove pour utiliser des prises 433mhz avec arduino :

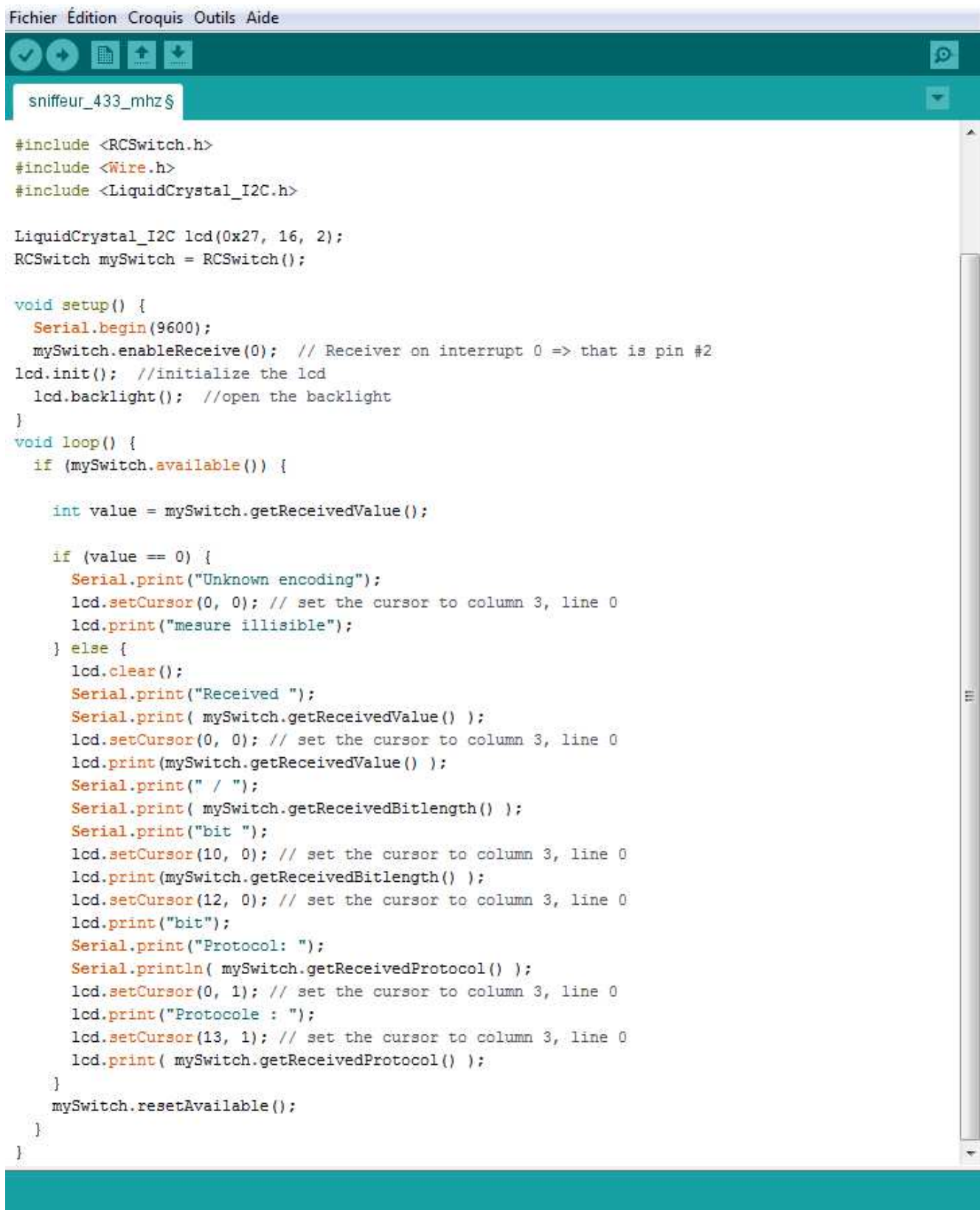
Les prises commandées qui utilisent les signaux 433mhz sont des équipements vraiment pratique et facile à détourner lorsqu'on souhaite les contrôler avec un arduino pour faire de la domotique à 2 €. A noter tout de même qu'il existe les signaux 433mhz cryptés et les signaux 433mhz non crypté. Le 433 megahertz est une fréquence répandue que l'on retrouve souvent dans nos habitations. Porte de garage, volet roulant télécommandé, portail électrique, store commandé à distance ou encore sonnette sans fil sont des équipements que l'on ne souhaite pas voir se mettre en action tout seul car un petit malin à capté leurs signaux et s'amuse à les émettre. Ces équipements utilisent donc des signaux 433mhz cryptés. Nous allons voir ici comment contrôler des signaux 433mhz non cryptés tels que ceux qui se trouvent dans les prises commandées sans fil.

Il suffit pour cela de ce munir d'un arduino, d'un récepteur 433mhz pour arduino et d'un écran LCD puis d'effectuer le montage suivant :



L'écran LCD est raccordé à l'arduino en utilisant le protocole I2C. Le fil signal du récepteur 433mhz est raccordé au pin Digital N°2 de l'arduino. Les bornes +5V et GND sont raccordées aux bornes correspondantes de l'arduino.

Voici maintenant le code à introduire dans l'arduino pour recevoir les signaux :



```
Fichier Édition Croquis Outils Aide
sniffeur_433_mhz $

#include <RCSwitch.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
RCSwitch mySwitch = RCSwitch();

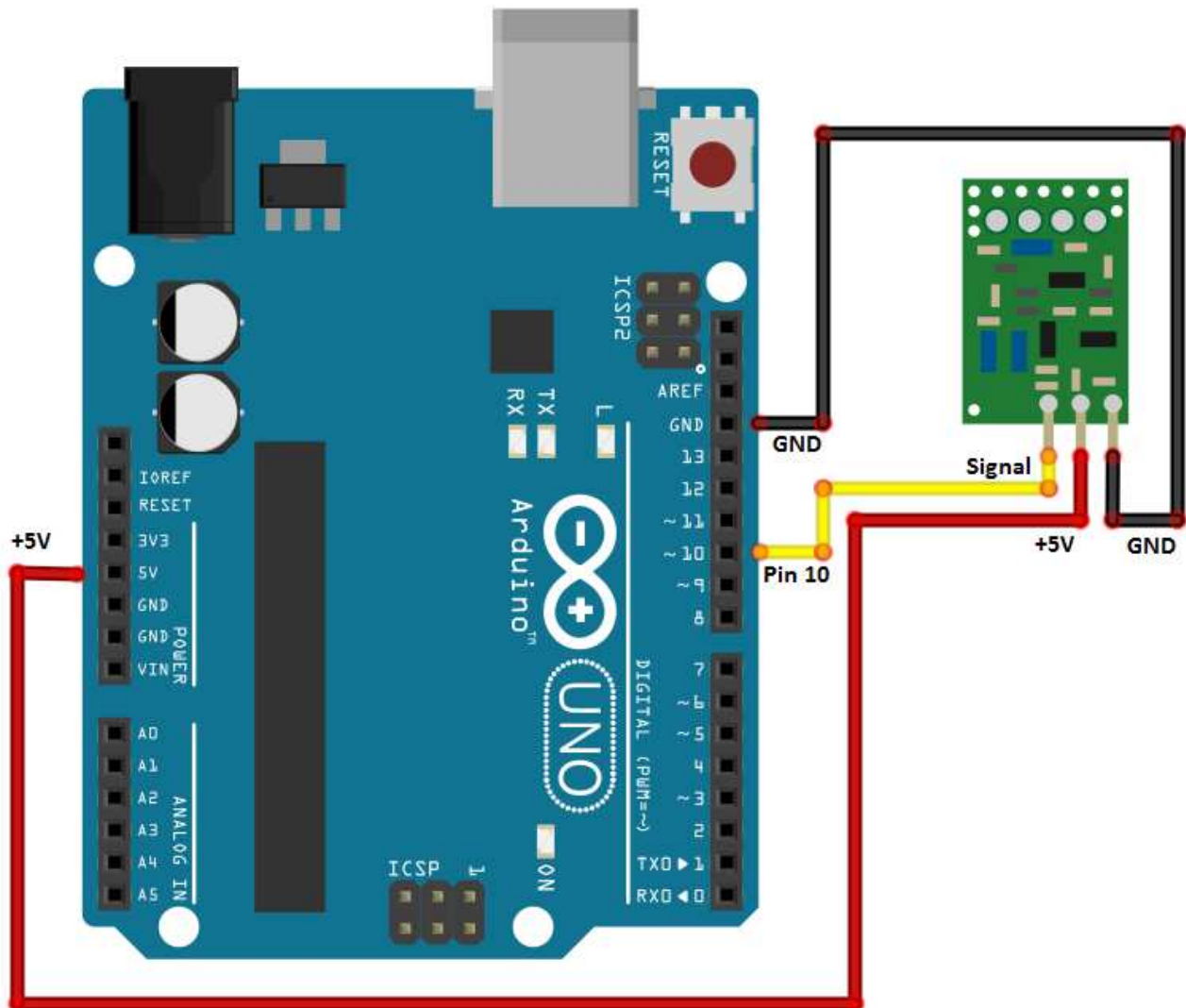
void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Receiver on interrupt 0 => that is pin #2
  lcd.init(); //initialize the lcd
  lcd.backlight(); //open the backlight
}
void loop() {
  if (mySwitch.available()) {

    int value = mySwitch.getReceivedValue();

    if (value == 0) {
      Serial.print("Unknown encoding");
      lcd.setCursor(0, 0); // set the cursor to column 3, line 0
      lcd.print("mesure illisible");
    } else {
      lcd.clear();
      Serial.print("Received ");
      Serial.print( mySwitch.getReceivedValue() );
      lcd.setCursor(0, 0); // set the cursor to column 3, line 0
      lcd.print(mySwitch.getReceivedValue() );
      Serial.print(" / ");
      Serial.print( mySwitch.getReceivedBitlength() );
      Serial.print("bit ");
      lcd.setCursor(10, 0); // set the cursor to column 3, line 0
      lcd.print(mySwitch.getReceivedBitlength() );
      lcd.setCursor(12, 0); // set the cursor to column 3, line 0
      lcd.print("bit");
      Serial.print("Protocol: ");
      Serial.println( mySwitch.getReceivedProtocol() );
      lcd.setCursor(0, 1); // set the cursor to column 3, line 0
      lcd.print("Protocole : ");
      lcd.setCursor(13, 1); // set the cursor to column 3, line 0
      lcd.print( mySwitch.getReceivedProtocol() );
    }
    mySwitch.resetAvailable();
  }
}
```

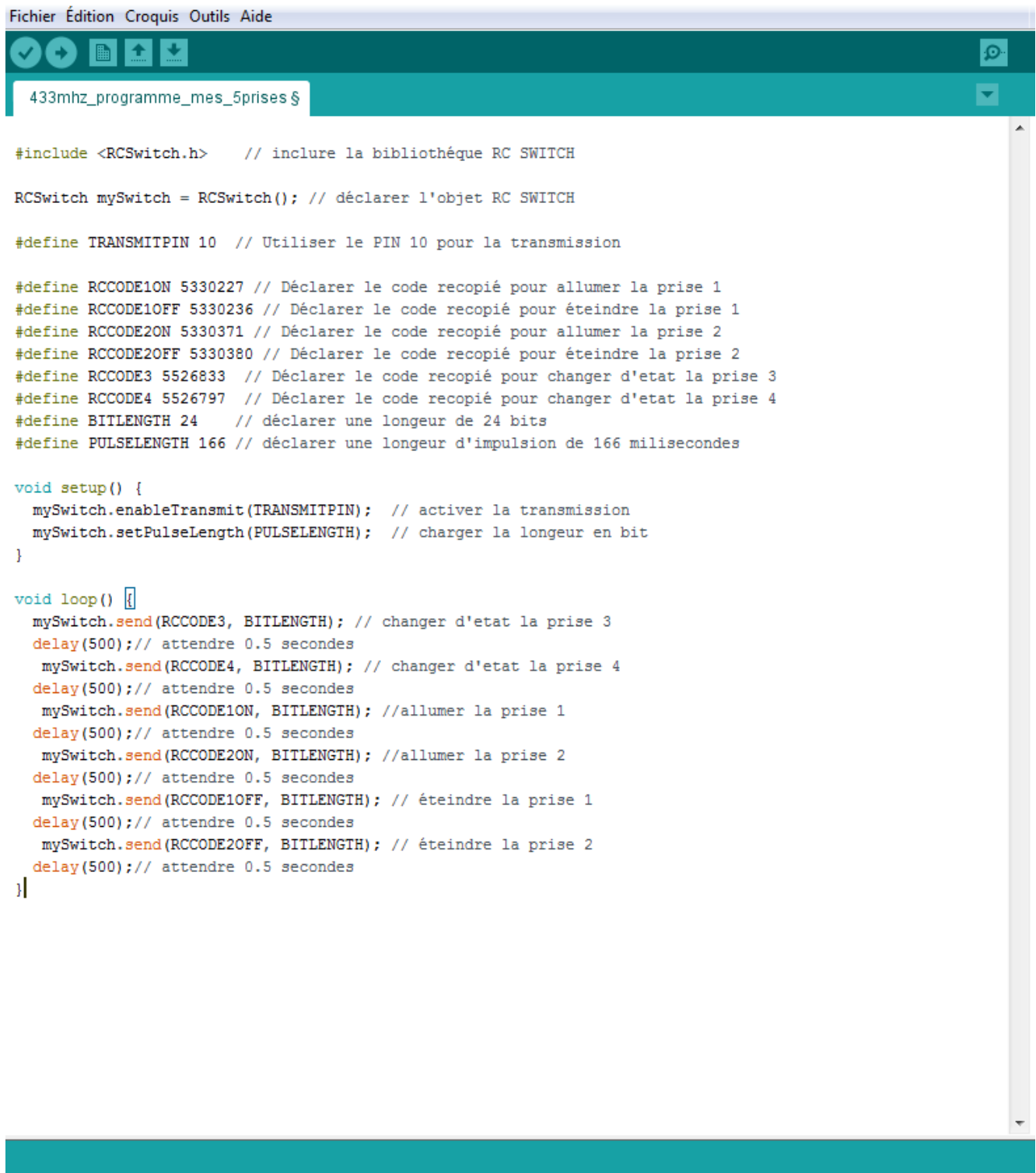
Il suffit alors de le téléverser dans l'arduino, et de lire les signaux non cryptés en appuyant sur les boutons des télécommandes émettrices.

Une fois les informations récupérées, avec un autre arduino, et un module émetteur, il faut effectuer ce montage pour qu'arduino puisse remplacer les télécommandes :



La borne signal de l'émetteur est raccordée au pin Digital 10 de l'arduino UNO, +5V et GND sont raccordés aux bornes correspondantes de l'arduino.

Maintenant, il faut recopier le code suivant pour commander les télécommandes. En remplaçant tout de même vos valeurs de signaux à la place de celles qui sont écrites dans le programme :



```
Fichier Édition Croquis Outils Aide
433mhz_programme_mes_5prises $

#include <RCSwitch.h> // inclure la bibliothèque RC SWITCH

RCSwitch mySwitch = RCSwitch(); // déclarer l'objet RC SWITCH

#define TRANSMITPIN 10 // Utiliser le PIN 10 pour la transmission

#define RCCODE1ON 5330227 // Déclarer le code recopié pour allumer la prise 1
#define RCCODE1OFF 5330236 // Déclarer le code recopié pour éteindre la prise 1
#define RCCODE2ON 5330371 // Déclarer le code recopié pour allumer la prise 2
#define RCCODE2OFF 5330380 // Déclarer le code recopié pour éteindre la prise 2
#define RCCODE3 5526833 // Déclarer le code recopié pour changer d'état la prise 3
#define RCCODE4 5526797 // Déclarer le code recopié pour changer d'état la prise 4
#define BITLENGTH 24 // déclarer une longueur de 24 bits
#define PULSELENGTH 166 // déclarer une longueur d'impulsion de 166 millisecondes

void setup() {
  mySwitch.enableTransmit(TRANSMITPIN); // activer la transmission
  mySwitch.setPulseLength(PULSELENGTH); // charger la longueur en bit
}

void loop() {
  mySwitch.send(RCCODE3, BITLENGTH); // changer d'état la prise 3
  delay(500); // attendre 0.5 secondes
  mySwitch.send(RCCODE4, BITLENGTH); // changer d'état la prise 4
  delay(500); // attendre 0.5 secondes
  mySwitch.send(RCCODE1ON, BITLENGTH); // allumer la prise 1
  delay(500); // attendre 0.5 secondes
  mySwitch.send(RCCODE2ON, BITLENGTH); // allumer la prise 2
  delay(500); // attendre 0.5 secondes
  mySwitch.send(RCCODE1OFF, BITLENGTH); // éteindre la prise 1
  delay(500); // attendre 0.5 secondes
  mySwitch.send(RCCODE2OFF, BITLENGTH); // éteindre la prise 2
  delay(500); // attendre 0.5 secondes
}
```

En téléversant ce programme dans l'arduino vous êtes maintenant en mesure de commander les prises dont vous avez recopié les signaux. A noter qu'une petite antenne soudée sur l'émetteur augmente la portée de celui-ci.