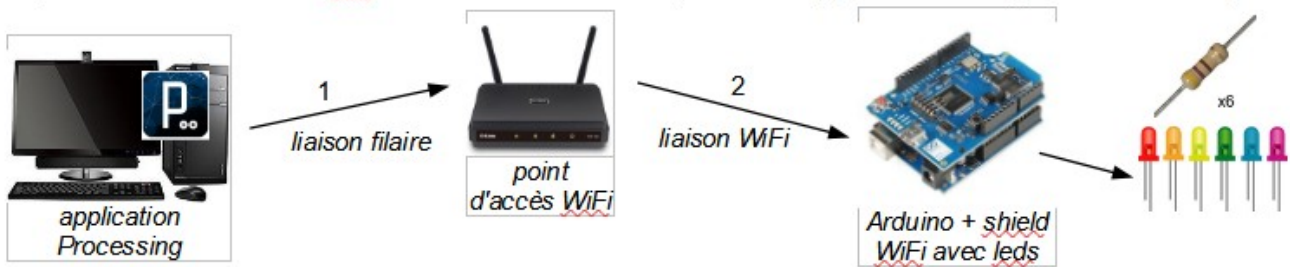


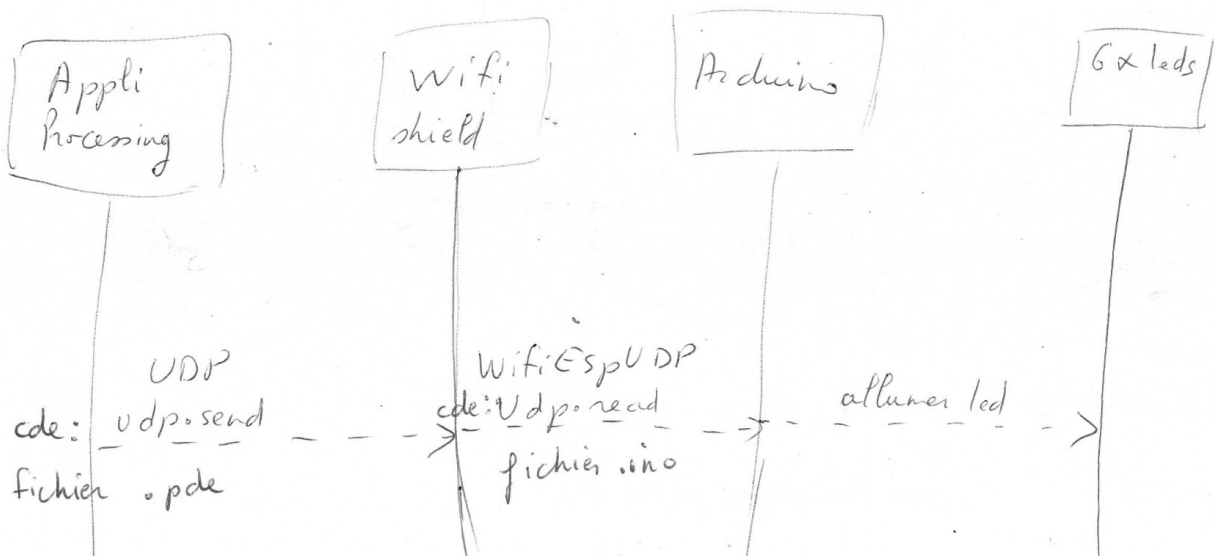
Commander des leds à partir d'une appli Processing

L'objectif est de commander 6 leds connectées à l'Arduino à partir d'une application développée sur Processing.



Pour transmettre les données de l'application vers la maquette nous utiliserons le protocole UDP.

Diagramme de séquences :



Application Processing

```
/*
  Envoi d'un tableau de 6 entiers afin de commander
  6 leds
  utilisation du protocole UDP
  Ch. Le Bris
  janvier 2015
  */
import hypermedia.net.*

UDP udp;
String ip="192.168.11.XX";
int port=8888;
int tab[]={ 0, 0, 0, 0, 0, 0 };

void setup() {
  udp=new UDP(this, 6000);
  udp.log(true); // affichage de messages dans la
  console
  size(600, 400);
  background(255);
  smooth();
}

void draw() {
  background(255);
  text("COMMANDE DES SPOTS", 220, 20);
  for (int i=5;i>=0;i--) {
    fill(0);
    text("LED"+i, 85+80*(5-i), 55);
    if (tab[i]==1) fill(#00FF00);
    else fill(#FF0000);
    ellipse(100+80*(5-i), 100, 60, 60);
  }
  fill(0);
  text("OFF", 40, 230);
  text("ON", 40, 265);
  for (int i=5;i>=0;i--) {
    text("I"+i, 95+80*(5-i), 180);
    noFill();
    rect(80+80*(5-i), 200, 40, 80);
    fill(0);
    if (tab[i]==0)
      rect(80+80*(5-i), 200, 40, 40);
    else
      rect(80+80*(5-i), 240, 40, 40);
  }
}

void mousePressed()
{
  for (int i=5;i>=0;i--) {
    if ((mouseX>(80+80*(5-i)))&&(mouseX<(120+80*(5-
i)))&&(mouseY>200)&&(mouseY<280)) tab[i]=1-tab[i];
  }
  udp.send("#"+str(tab[0])+str(tab[1])+str(tab[2])
+str(tab[3])+str(tab[4])+str(tab[5]), ip, port);
}
}
```

@IP de votre shield
WiFi

Arduino

```
/*
  Réception d'un tableau de 6 entiers afin de
  commander 6 leds
  utilisation du protocole UDP
  Ch. Le Bris
  janvier 2015
  */
#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>

int status = WL_IDLE_STATUS;
char ssid[] = "STI2D_SIN"; // your network SSID
(name)
char pass[] = "sti2dsin"; // your network
password

unsigned int localPort = 8888; // local port
to listen on
char packetBuffer[255]; //buffer to hold incoming
packet

WiFiUDP Udp;

void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  // déclaration en sortie des broches où sont
  cablées les leds
  pinMode(2,OUTPUT);

  // attempt to connect to Wifi network:
  while ( status != WL_CONNECTED) {
    Serial.print("Connexion au reseau : ");
    Serial.println(ssid);
    status = WiFi.begin(ssid,pass);
    // wait 5 seconds for connection:
    delay(5000);
  }
  Serial.println("Connecte au WiFi");
  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("Adresse IP : ");
  Serial.println(ip);

  Serial.println("\nPrete a recevoir des
donnees...");
  Udp.begin(localPort);
}

void loop() {
  int packetSize = Udp.parsePacket();
  if (packetSize)
  {
    Serial.print("Recu ");
    Serial.print(packetSize);
    Serial.print(" octets de ");
    IPAddress remoteIp = Udp.remoteIP();
    Serial.print(remoteIp);
    Serial.print(":");
    Serial.println(Udp.remotePort());

    // read the packet into packetBuffer
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) packetBuffer[len] = 0;
    Serial.println("Contenu:");
    Serial.println(packetBuffer);

    // envoi des états logiques sur les leds
    Udp.read(packetBuffer,255);
    if (packetBuffer[0]==0x23) {
      digitalWrite(2,packetBuffer[1]-48);
    }
  }
}
}
```

A compléter pour les autres
leds