

TouchOSC

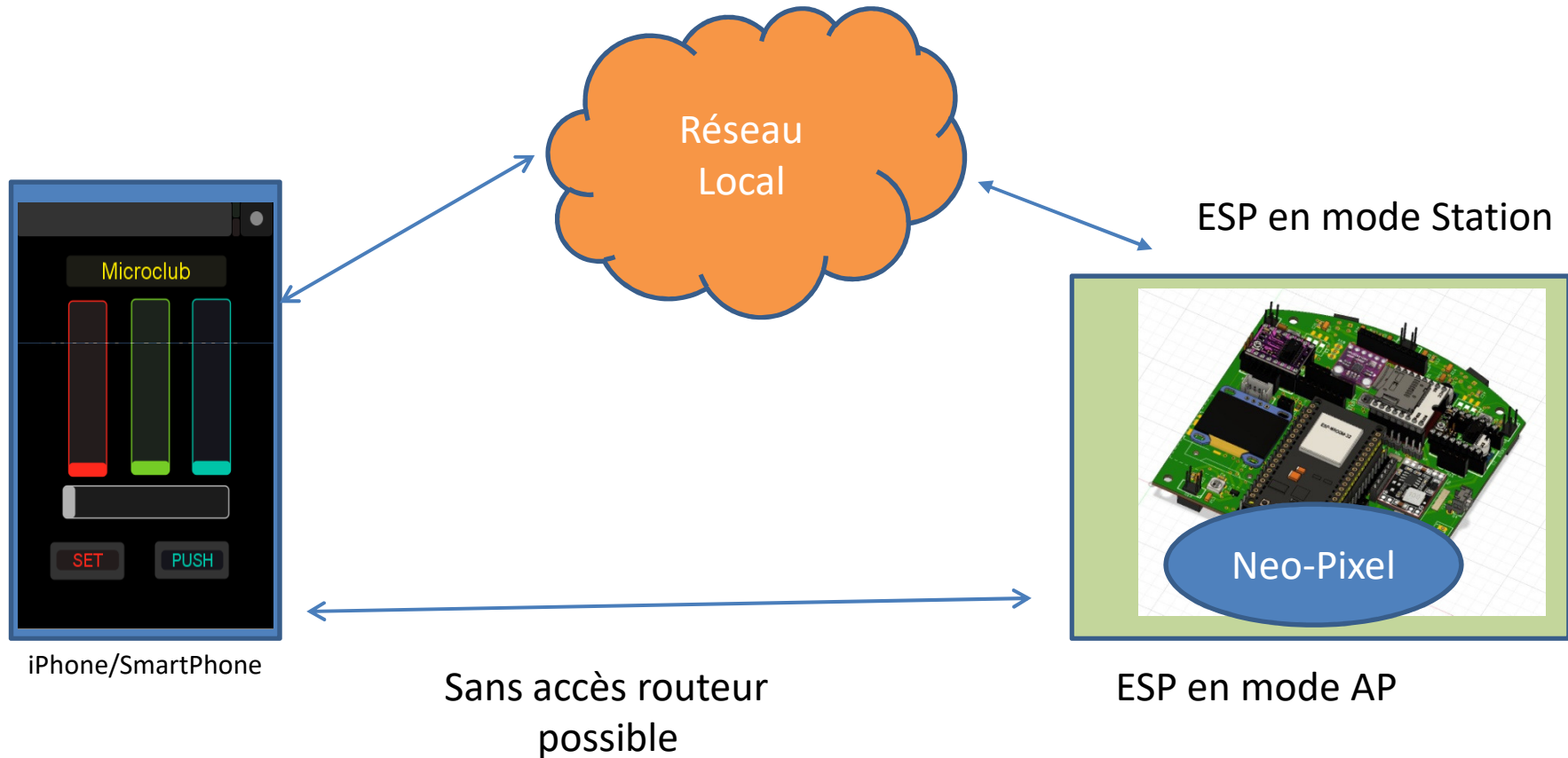
Interface ESP-tel.portable facile

Rolf Ziegler Avril 2019

Agenda

- Principe de fonctionnement
- Communication UDP
- Configuration interface pour le téléphone
- Librairie et code côté ESP (8266 ou 32)
- Démonstration

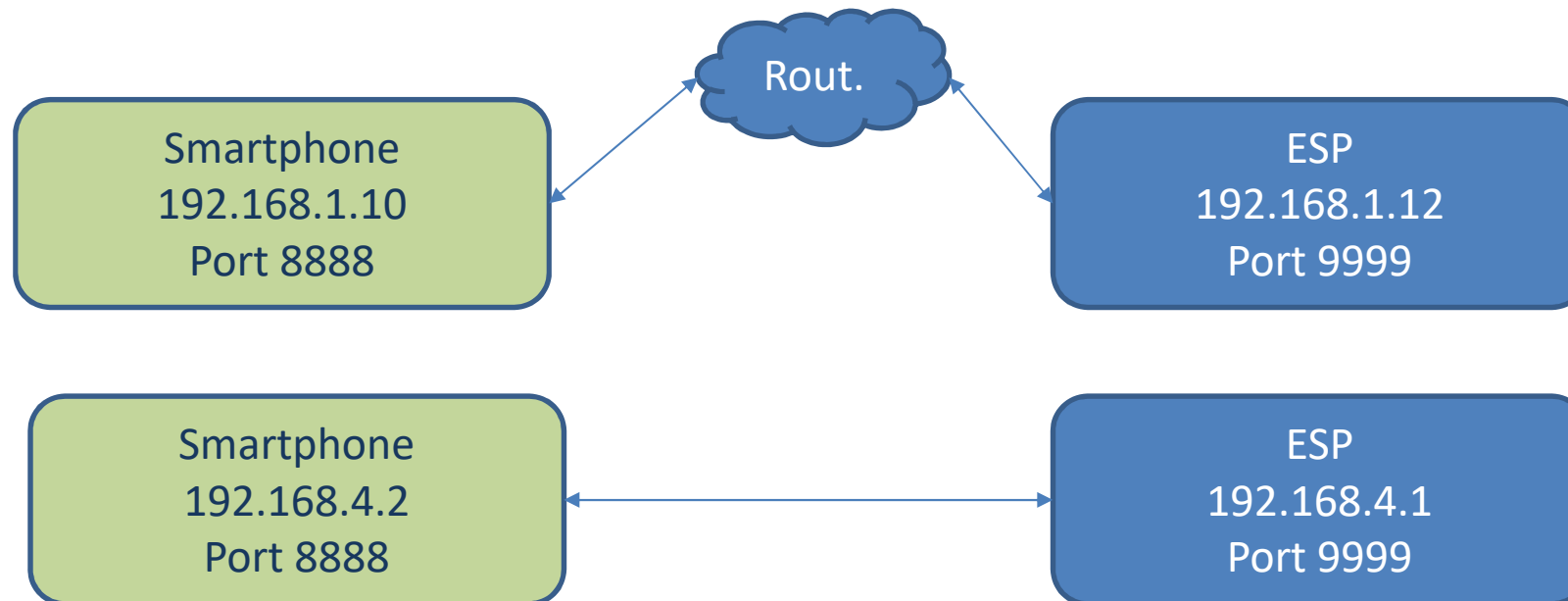
Principe de fonctionnement



TouchOSC

Communication UDP

- UDP est une alternative à TCP-IP, Protocol réduit, plus rapide
- On peut communiquer entre 2 composants avec leur adresse IP + un port de communication

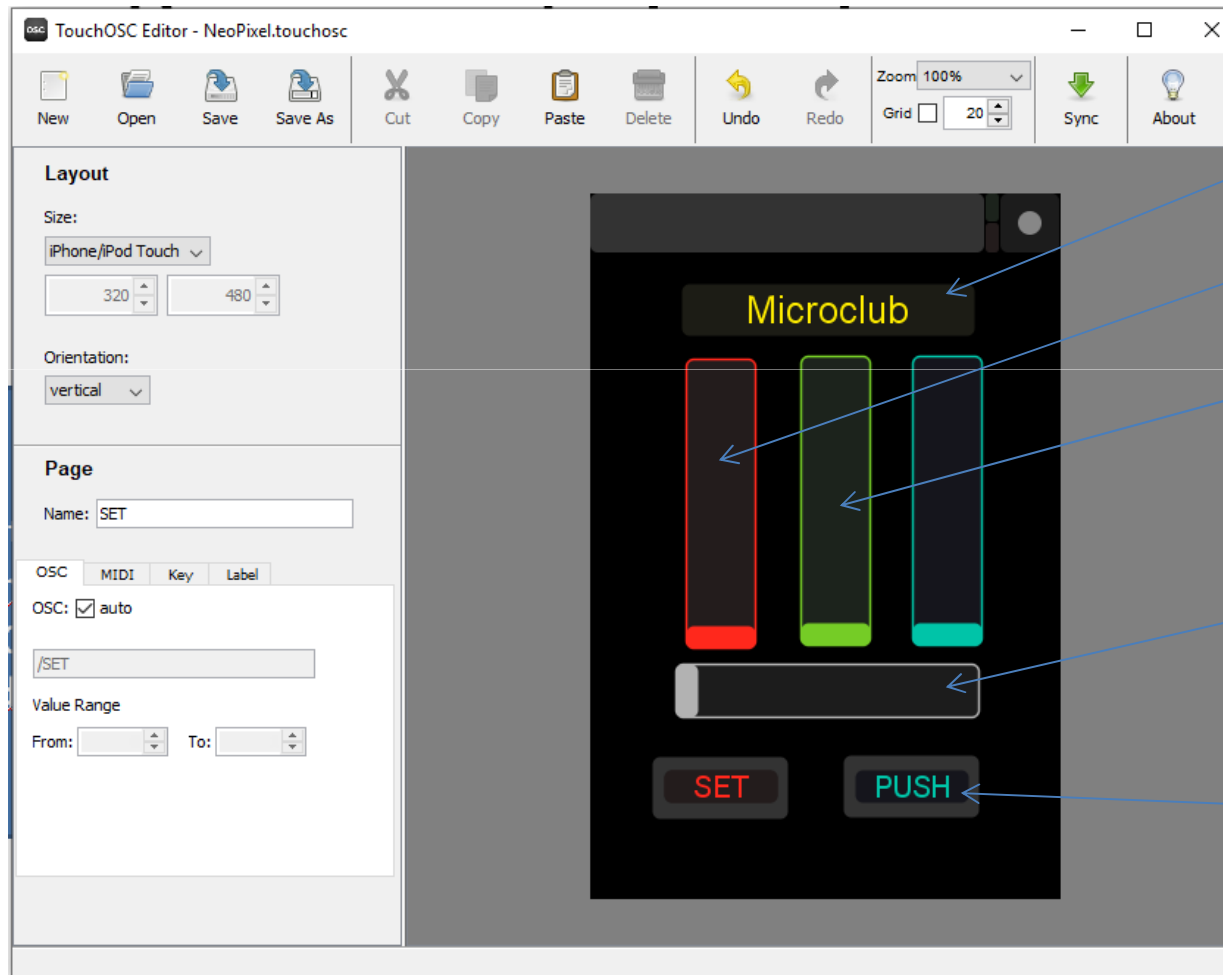


TCP comparé à UDP

- HTTP, qui sert à accéder aux sites internet (autrement dit : le web)
 - FTP, qui sert à échanger des fichiers entre 2 ordinateurs
 - POP3 et IMAP qui sert à lire ses emails
 - SMTP qui sert quant à lui à envoyer des emails
 - TCP Protocol de communication (Transmission Communication Protocol)
 - UDP Protocol utilisateur (User Datagram Protocol)
-
- TCP incluse un accusé de réception, UDP pas
 - En communication, l'information est divisée en paquets d'au moins 1500 octets. On considère que 10% des paquets sont perdus ou mal transmis.
 - TCP permet de demander de renvoyer des paquets défectueux ou manquant, UDP pas.
 - UDP est par ces faits beaucoup plus rapide que TCP
 - Si l'on envoie par exemple une position ou température toutes les 250ms, il n'est pas important si un paquet est perdu, il sera compensé/corrigé par le prochain paquet.

TouchOSC

Configuration côté applic.



- Objets
 - /1/logo
 - /1/rouge
 - /1/vert
 - /1/bleu
 - /1/intens
 - /1/set
 - /1/push

TouchOSC

Téléchargement

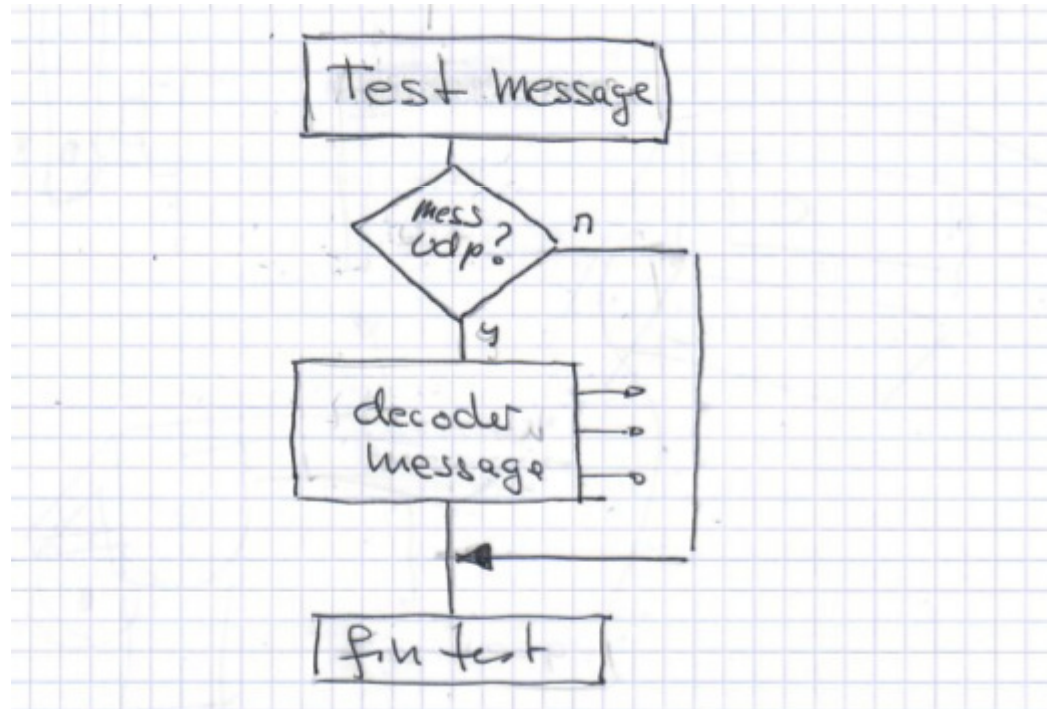
- Une fois l'interface créée on doit la télécharger
- Sélectionner « Sync »
- Connecter le smartphone sur le même réseau
- ! Sur le même segment (même SSID)
- Dans le menu on trouve ADD
- L'interface est télécharger sur le smartphone
- iPhone ou Android (les 2 sont possible)
- !Noter tous les noms des objets pour la suite !

Envoi d'informations

- Les fonctions « message » envoient une (1) information à la fois
- Les fonctions « bundle » envoient plusieurs informations à la fois/par appel.
- Chaque message envoie à un destinataire identifié par son adresse IP et son port.

Réception d'informations

- Couverts par la librairie TouchOSC



! On utilise la fonction bundle: `bundle.dispatch("/1/intens", intensite);`
Pour décoder le message, donc pas besoins de faire du code pour comparer !

Librairie et code(1)

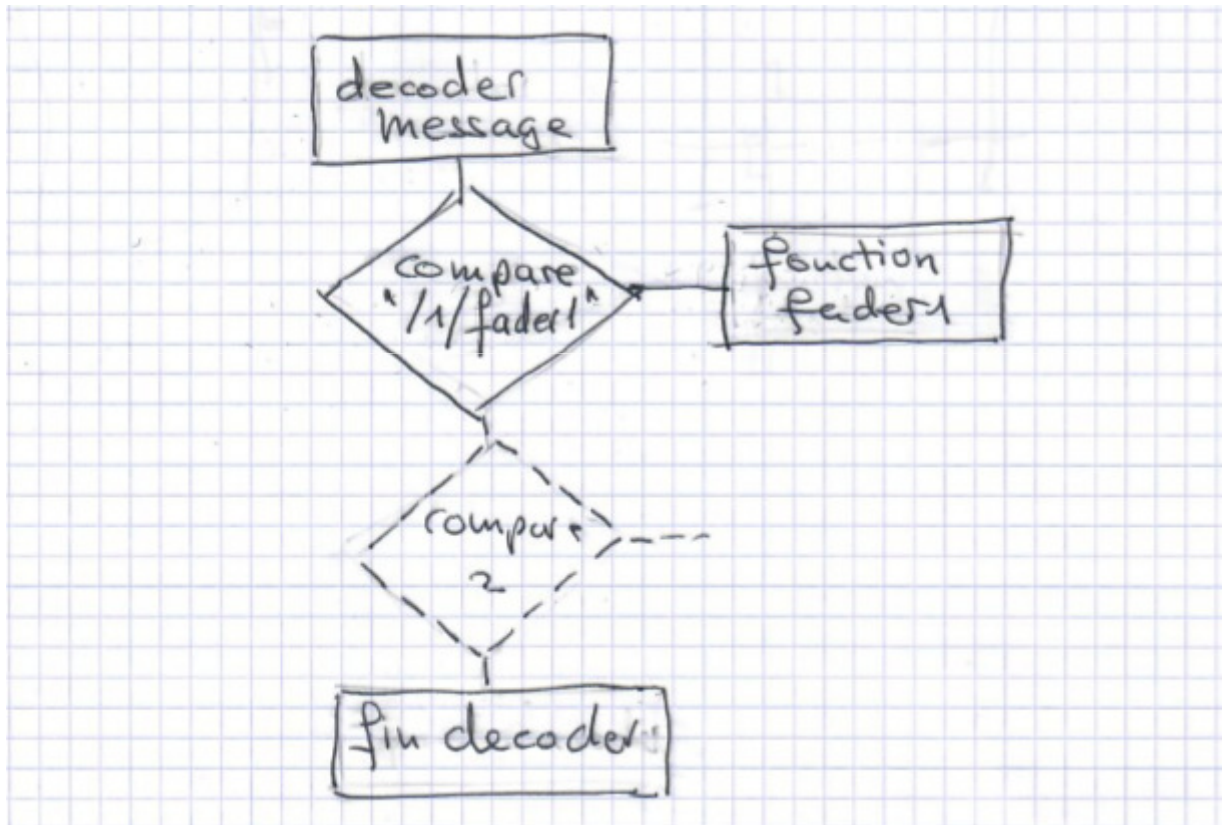
Réception de données/messages

- Des routines dans OSCbundle.h permettent de gérer la communication.

```
void loop() {
  OSCBundle bundle; // ouvrir une instance bundle
  int size = Udp.parsePacket(); // demander la longueur des données reçues
  if (size > 0) { // si bytes reçus >0
    while (size-- > 0) { //tant que compteur pas à zéro
      bundle.fill(Udp.read()); // copier les bytes dans le buffer bundle
    }
  }
  if (!bundle.hasError()) { // si pas d'erreur de transmission
    bundle.dispatch("/1/button1", fonction1); // on appelle fonct. bouton1 si pressé
  } else {
    error = bundle.getError(); // si non il y a une erreur, ignorer ou indiquer.
    Serial.print("error: ");
    Serial.println(error);
  }
}
```

Librairie et code

Récupérer une valeur:



! On utilise la fonction bundle: `bundle.dispatch("/1/intens", intensite);`
Pour décoder le message, donc pas besoins de faire du code pour comparer !

Librairie et code

Récupérer une valeur:

```
void intensite(OSCMessage &msg) {  
  Serial.print("Intensity: ");  
  intensity = (uint8_t)msg.getFloat(0);  
  Serial.println(intensity);  
  
  FastLED.setBrightness( intensity );  
  FastLED.show();  
}
```

Librairie et code(1) un message à la fois

- Envoi de données:

```
// indiquer la librairie #include « OSCmessage.h »  
// connecter au routeur  
// outIp = ip de destination, outPort=Port de destination  
WiFiUDP Udp; // ouvrir une instance UDP pour envoyer et recevoir des  
    messages  
// préparer l'adresse du message à envoyer  
OSCMessage msg("/1/logo"); // texte sera affiché sur l'élément /1/logo  
msg.add("Online"); // on y indique que nous sommes online  
Udp.beginPacket(outIp, outPort);  
msg.send(Udp);  
Udp.endPacket();  
msg.empty();  
delay(500);
```

Librairie et code(2)

plusieurs messages à la fois

- Envoi de données:

```
// indiquer la librairie #include « OSCBundle.h »
```

```
// send à Bundle
```

```
OSCBundle budl;
```

```
// bundle.add permet d'envoyer l'adresse et la valeur en même temps
```

```
// suivant on envoie 2 valeurs d'une entrée analogique vers OSC
```

```
bndl.add("/1/curseur1".add(int32_t)analogRead(0));
```

```
bndl.add("/1/curseur2".add(int32_t)analogRead(2));
```

- Udp.beginPacket(outIp, outPort);
- msgOUT.send(Udp);
- delay(1);
- Udp.endPacket();
- msgOUT.empty();

Où trouver l'application et les bibliothèques

- Le site Hexler, chercher OSCtouch
- Ou directement sous

<https://hexler.net/software/touchosc>

- ! Le module ESP et le téléphone doivent être sur le même réseau du routeur (plage d'adresses IP)
- ! Même chose pour télécharger l'interface

Application et librairies

- Site de OSCTOUCH
 - Site: <https://hexler.net/>
 - Doc: <https://hexler.net/docs/touchosc>