



# Dispositif de transmission

CREATION D'UN MICROPROSSEUR

Chadi Manglé | SAE 13 | 1<sup>er</sup> Décembre 2022



## Remerciement

Je remercie M. Lee de nous avoir fourni les connaissances ainsi que le matériel afin de réaliser ce projet.

## Table des matières

Remerciement .....	2
Introduction .....	4
Compte rendu du TP <sub>3</sub> : .....	7
Compte rendu du TP <sub>4</sub> : .....	12
Conclusion : .....	21

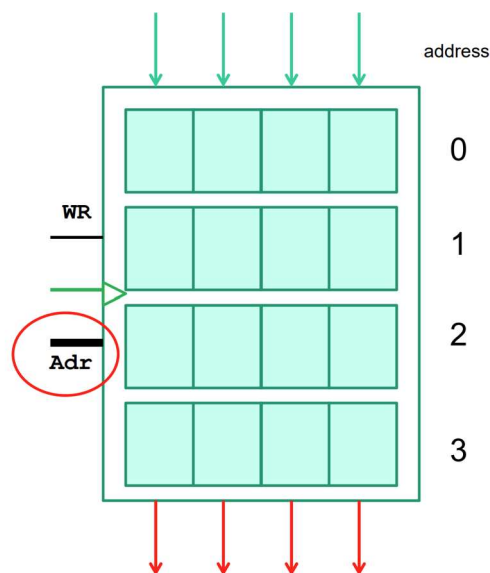
## Introduction

Lors de cette SAE nous devons réaliser un microprocesseur en Arduino.

Le microprocesseur est un circuit intégré complexe. Il résulte de l'intégration sur une puce de fonctions logiques combinatoires et séquentielles. Il est capable d'interpréter et d'exécuter les instructions d'un programme. Afin de faire ce programme nous avons préalablement réalisé en Arduino un banc de registre, un ALU, ainsi qu'un multiplexeur 2 vers 1 durant les TP1 et 2 du module R106.

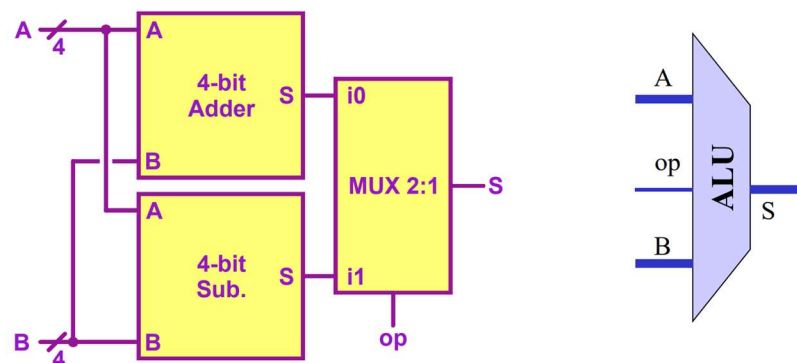
Un banc de registre est une mémoire interne au processeur dans laquelle sont rassemblés certains ou tous ses registres.

Schéma du banc de registre :



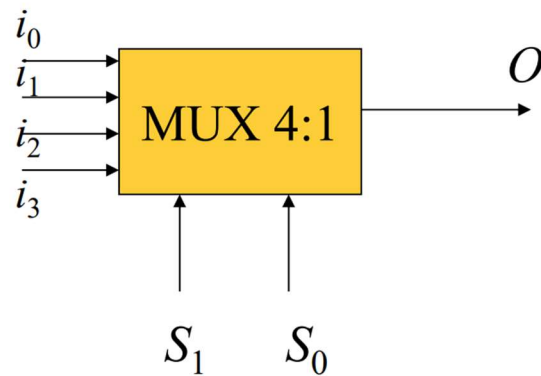
L'ALU (ou Unité arithmétique et logique en français) est un ensemble de circuits électroniques connectés logiquement de façon chargée d'effectuer des calculs arithmétiques ou logique.

Schéma d'un ALU<sub>2:1</sub> :



Un multiplexeur est un circuit permettant de concentrer sur une même voie de transmission différents types de liaisons en sélectionnant une entrée parmi N. Dans le TP nous allons utiliser un MUX<sub>4:1</sub>.

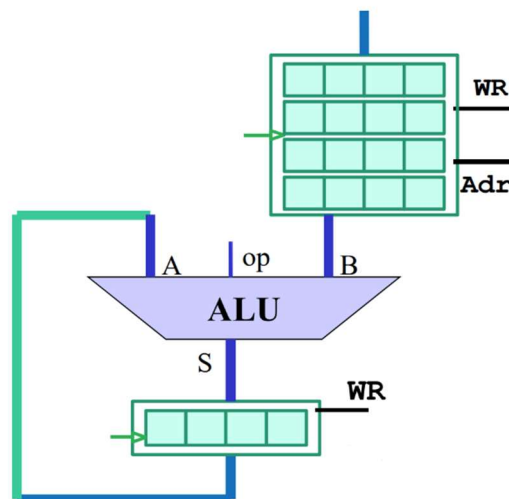
Schéma du multiplexeur :



En plus de ces quatre composants déjà possédés, nous devons créer à cette occasion un registre accumulateur.

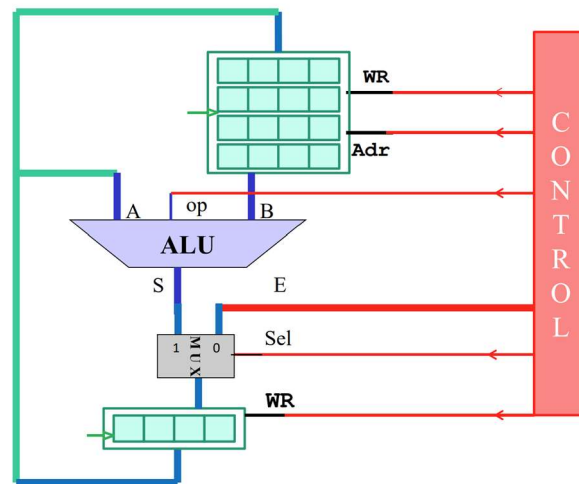
Un accumulateur est un registre spécial, incorporé dans certaines architectures de processeurs, où les résultats intermédiaires d'un UAL sont versés.

Schéma de l'accumulateur :



Tous ces composants sont utiles au bon fonctionnement du microprocesseur étant constitué de tous ces composants électroniques.

Schéma du microprocesseur :



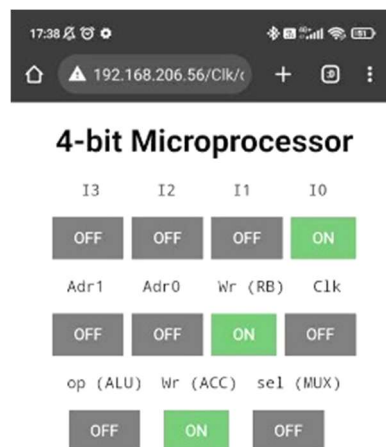
Dans le cadre de la réalisation de microprocesseur en Arduino, les contrôles seront directement faits depuis un téléphone mobile. Pour ce faire, nous allons utiliser des cartes ESP32. Les cartes ESP32 sont des cartes très polyvalentes orientées IoT (Internet of Things) qui intègrent la gestion du Wi-Fi.

## Compte rendu du TP3 :

Lors du TP3 on devait réaliser un microprocesseur 4 bits. La majorité des programmes intermédiaire étant déjà réalisés en amont, il fallait seulement les copier dans le fichier header « circuit.h ». Le travail restant est de coder l'accumulateur et de correctement affecter les variables de sortie.

Dans la consigne TP3 nous devions réaliser le calcul de  $7 + 3 - 6$  sur le microprocesseur. J'ai pris l'initiative de changer les valeurs des calculs en  $1 + 1 - 2$  pour m'assurer de la validité et du fonctionnement du microprocesseur et éviter les répétitions avec le compte rendu du TP4.

Pour effectuer ce calcul il faut tout d'abord, initialiser la valeur 0001 (1 en binaire) à l'adresse 00 du banc de registre.

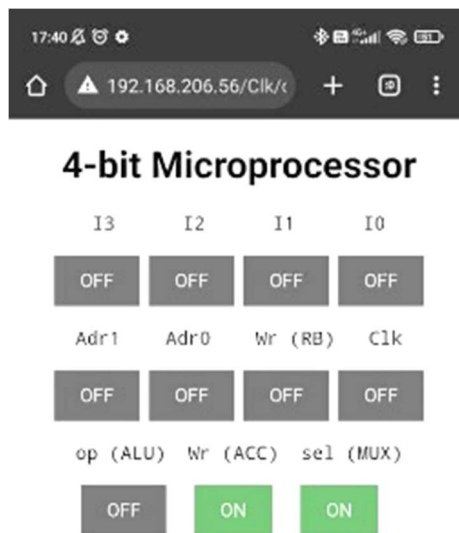


```
--- Register bank ---
Reg0[3..0]: 0001
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0000
```

Affichage de l'action réalisée dans le moniteur série.



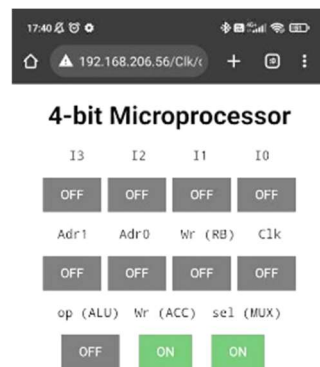
Ensuite on prend contenu et le contenu stocké à l'adresse 00 du banc de registre (0001) et on l'ajoute à l'accumulateur (0000).



```
--- Register bank ---
Reg0[3..0]: 0001
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0001
```

Affichage de l'action réalisée dans le moniteur série.

On refaire l'addition entre le de l'accumulateur (0001) et le banc de registre à l'adresse 00 (0001).



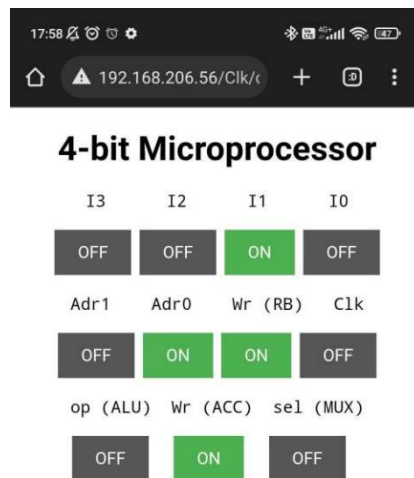
```
--- Register bank ---
Reg0[3..0]: 0001
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0010
Client disconnected.
```

Affichage de l'action réalisée dans le moniteur série.



Les LEDs affiche le contenu de l'accumulateur.

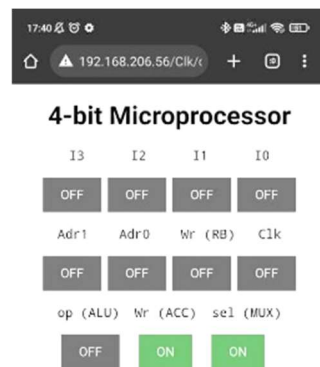
Ensuite on initialise la valeur 0010 (2 en binaire) à l'adresse 01 du banc de registre.



```
--- Register bank ---
Reg0[3..0]: 0001
Reg1[3..0]: 0010
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0010
```

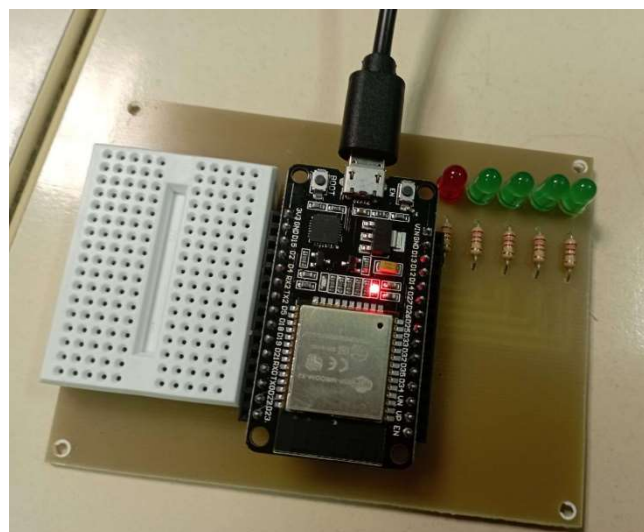
Affichage de l'action réalisée dans le moniteur série.

Puis pour terminer le calcul on soustrait la valeur à l'adresse 01 (0002) avec celle de l'accumulateur (0002)



```
--- Register bank ---  
Reg0[3..0]: 0001  
Reg1[3..0]: 0010  
Reg2[3..0]: 0000  
Reg3[3..0]: 0000  
--- Accumulator ---  
ACC[3..0]: 0000
```

Affichage de l'action réalisée dans le moniteur série.



L'accumulateur est égal à 0000 donc toutes les LEDs sont éteintes.

## Compte rendu du TP4 :

Pour le TP4 nous devons aussi faire un microprocesseur à la différence que l'on devrait le contrôler directement à partir des instructions binaires.

Instruction	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
addition	0	0	Adresse		x	x
soustraction	0	1	adresse		x	x
copie accumulateur – banc de registres	1	0	adresse		x	x
chargement accumulateur	1	1	nombre à charger			

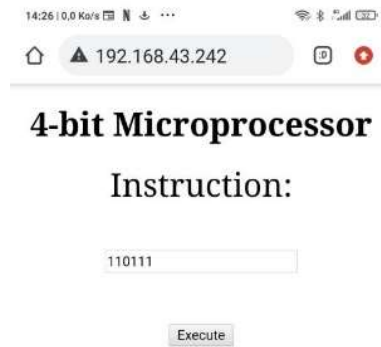
Il faut donc assigner à chaque instruction son expression logique comme vu lors du TD2 du module R1o6.

$$\begin{aligned} \text{address} &= b2.b3 \\ \text{operation} &= b4 \\ \text{selector} &= \overline{b5} \\ W r_{ACC} &= \overline{b5}.\overline{b4} + \overline{b5}.b4 + b5.b4 \\ W r_{RB} &= b5.\overline{b4} \\ E_0 &= b3 + b2 + b1 + b0 \end{aligned}$$

Le reste des composants électroniques étant les mêmes que pour le TP3 on utilise exactement le même fichier de header « circuit.h ».

Dans le TP4 on nous demande faire le calcul de  $7 + 3 - 6$  avec le microprocesseur afin de vérifier la validité du programme.

Pour ce faire le il faut d'abord initialiser la valeur de l'accumulateur à 0111 (7 en binaire).



14:26 | 0.0 Ks/s | [Icons] | [Signal] [Battery]

192.168.43.242

## 4-bit Microprocessor

Instruction:

Execute

```
Instruction: 110111
I[3..0]: 0111
Wr_ACC: 1
Wr_RB: 0
Adr: 10
op: 1
sel: 0
--- Register bank ---
Reg0[3..0]: 0000
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0111
```

Affichage de l'action réalisée sur le moniteur série.

On copie ensuite cette valeur à l'adresse 00 du banc de registre.

15:18 | 1,9 Ko/s | 📶 | 🔋 | 100%

🏠 192.168.43.242 📄 📌

## 4-bit Microprocessor

Instruction:

■ ● ◀ ▶

```
Instruction: 100000
I[3..0]: 0000
Wr_ACC: 0
Wr_RB: 1
Adr: 00
op: 0
sel: 0
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0111
```

Affichage de l'action réalisée sur le moniteur série.

On change la valeur de l'accumulateur à 0011 (3 en binaire).



```
Instruction: 110011
I[3..0]: 0011
Wr_ACC: 1
Wr_RB: 0
Adr: 00
op: 1
sel: 0
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0000
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0011
```

Affichage de l'action réalisée sur le moniteur série.



On copie ensuite cette valeur à l'adresse 00 du banc de registre.

15:48 | 0,0 Ko/s | [Icons] | [Signal] [Battery]

🏠 192.168.43.242 [Icons]

---

## 4-bit Microprocessor

### Instruction:

100100

Execute

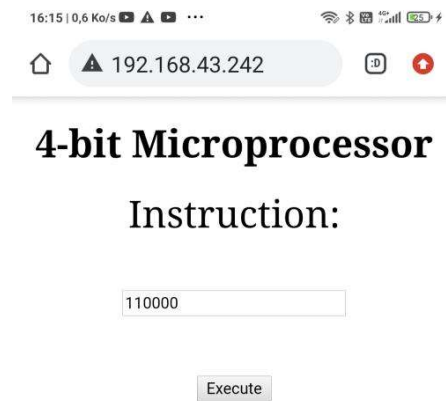
---

■ ● ◀ ▶

```
Instruction: 100100
I[3..0]: 0100
Wr_ACC: 0
Wr_RB: 1
Adr: 01
op: 0
sel: 0
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0011
Reg2[3..0]: 0000
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0011
```

Affichage de l'action réalisée sur le moniteur série.

On utilise le même procédé pour mettre 0110 (6 en binaire) à l'adresse 01 de l'accumulateur. Puis on réinitialise l'accumulateur en lui assignant la valeur 0000 (0 en binaire).



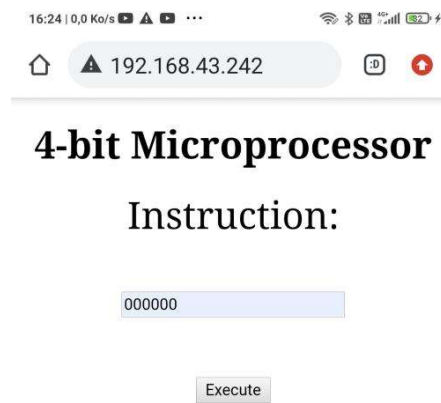
```

Instruction: 110000
I[3..0]: 0000
Wr_ACC: 1
Wr_RB: 0
Adr: 00
op: 1
sel: 0
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0011
Reg2[3..0]: 0110
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0000

```

Affichage de l'action réalisée sur le moniteur série.

Ensuite on additionne le contenu a l'adresse 00 banc de registre (0111) avec le contenu de l'accumulateur (0000) à l'adresse 00. On aurait pu optimiser le processus en mettant directement 0111 (7 en binaire) dans l'accumulateur sans le stocké dans le banc de registre.



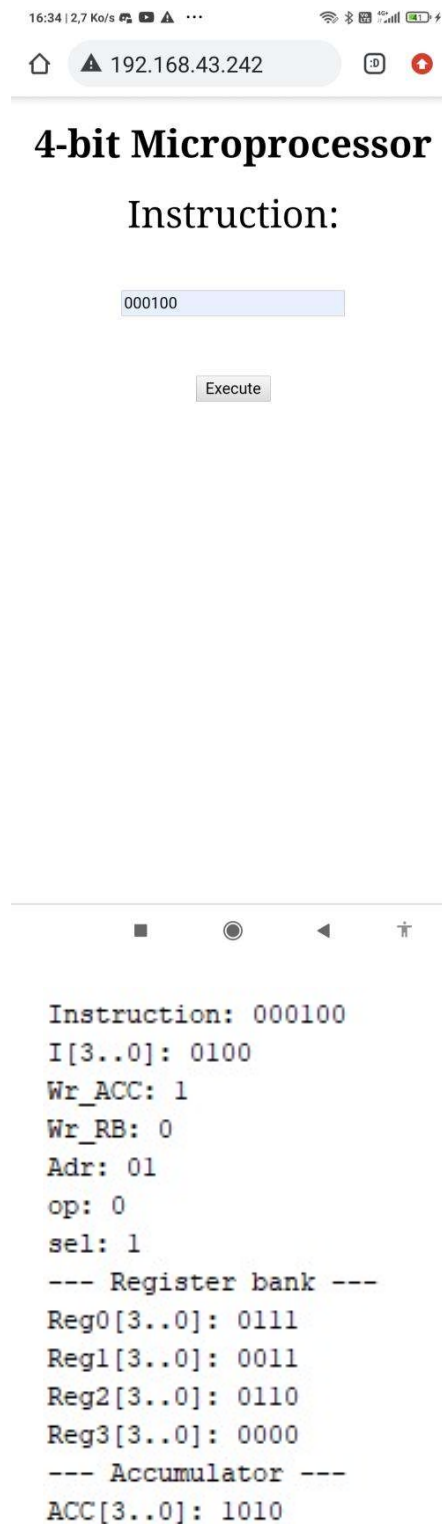
```

Instruction: 0000000
I[3..0]: 0000
Wr_ACC: 1
Wr_RB: 0
Adr: 00
op: 0
sel: 1
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0011
Reg2[3..0]: 0110
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0111

```

Affichage de l'action réalisée sur le moniteur série.

On fait après l'addition de l'accumulateur avec le contenu du banc de registre à l'adresse 01.



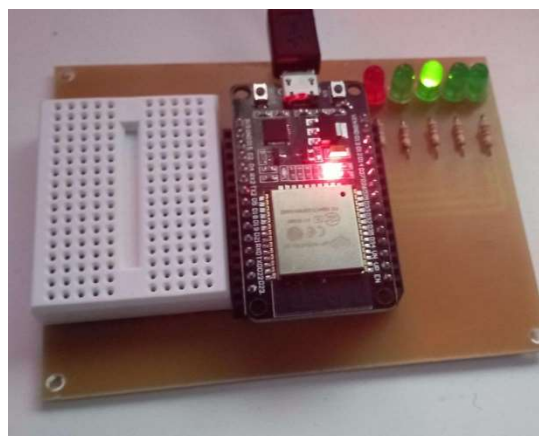
Affichage de l'action réalisée sur le moniteur série.

Et pour finir on fait la soustraction de l'accumulateur avec le contenu du banc de registre à l'adresse 10.



```
Instruction: 011000
I[3..0]: 1000
Wr_ACC: 1
Wr_RB: 0
Adr: 10
op: 1
sel: 1
--- Register bank ---
Reg0[3..0]: 0111
Reg1[3..0]: 0011
Reg2[3..0]: 0110
Reg3[3..0]: 0000
--- Accumulator ---
ACC[3..0]: 0100
```

Affichage de l'action réalisée sur le moniteur série.



Comme lors du TP3 les LEDs affiche le contenu de l'accumulateur.

## Conclusion :

Ces deux TP nous ont permis d'en savoir plus sur les microprocesseurs ainsi que sur l'architecture des systèmes numériques et la programmation en C++. De plus nous avons eu l'occasion de travaillé avec les cartes ESP32 et Arduino ce qui a permis de nous initier à l'IoT.

Ces quelques dernières années, l'IoT est devenu l'une des technologies les plus importantes et développé. Maintenant que nous pouvons connecter des objets du quotidien à Internet, des communications sont possibles en toute fluidité entre les personnes, les processus et les objets.

Dans le futur nous pouvons imaginer un monde où la domotique se trouvera au cœur de nos vies. La domotique opère dans un champ technique et informatique très vaste. Elle permet de contrôler la plupart des appareils et dispositifs électriques de la maison, de l'éclairage et le chauffage jusqu'aux équipements audiovisuels et électroménagers, en passant par l'ouverture des fenêtres.

Etant donné que la formation Réseau et Télécommunication a pour but de nous enseigner l'administration des réseaux et comment assurer leur sécurité, il est donc normal que la maîtrise de l'IoT fasse partie de nos compétences car l'entretien et la sécurisation de ces réseaux pourrait être central dans notre vie professionnelle.