

SAE301 – Mettre en œuvre un système de transmission

1) Présentation du projet

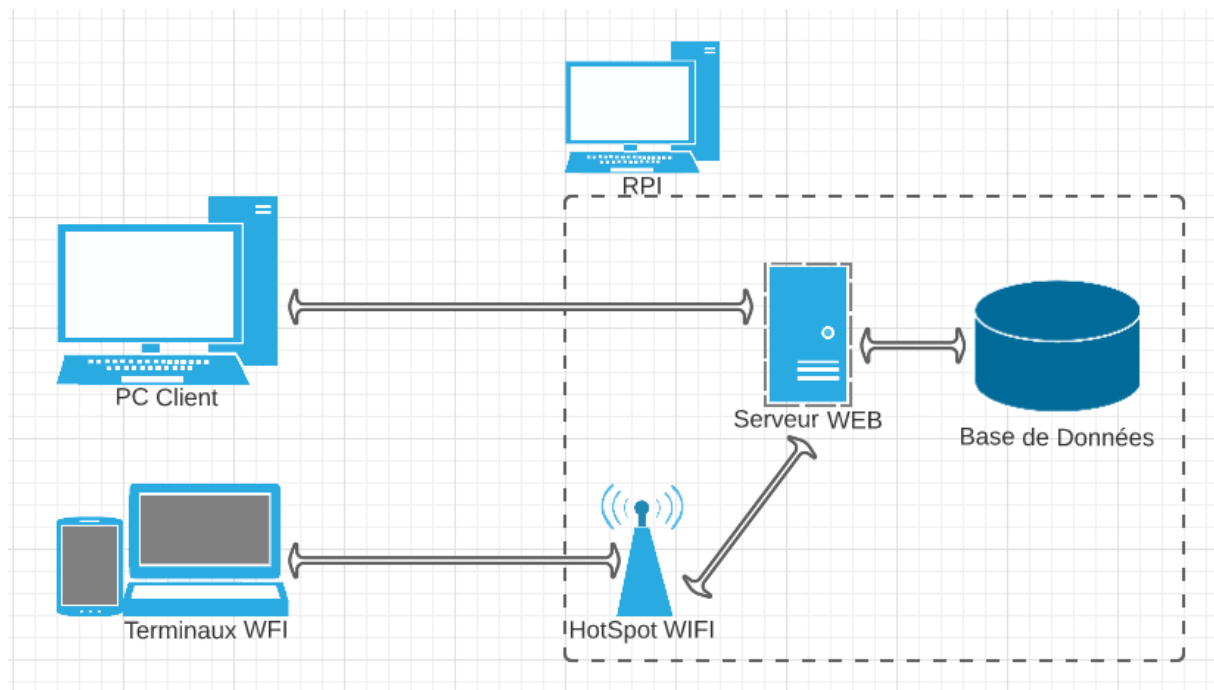


Figure 1 : Schéma infrastructure réseau

Sur le RPI est hébergé :

- Serveur WEB
- Serveur SSH
- Serveur VNC
- Serveur SIP
- Serveur DHCP
- Base de données
- Module zigbee
- Fail2Ban
- HotSpot WIFI

2) Création de la BDD

La base de données est gérée avec PHPmyAdmin sur le RPI. Elle contient une base nommée SAE301. Dans cette base on y retrouve une table « mesures » qui contient les champs suivants :

- 1 champ "id" → contient les identifiants des enregistrements
- 1 champ "date" → contient l'horodatage des mesures
- 1 champ "Temp" → contient la température fournie par le serveur
- 1 champ "Vsolare" → contient la puissance lumineuse fournie par le serveur

- 1 champ "Pconso" → contient la puissance électrique consommée fournie par le serveur

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	id	int(11)			No	None		AUTO_INCREMENT
☐	2	date	datetime			No	None		
☐	3	Temp	int(11)			Yes	NULL		
☐	4	Vsolaire	int(11)			Yes	NULL		
☐	5	Pconso	int(11)			Yes	NULL		

Figure 2 : Capture de la structure de la BDD

☐	Edit	Copy	Delete	131	2023-01-23 10:19:27	1718	234	1445
☐	Edit	Copy	Delete	132	2023-01-23 10:20:01	1718	222	405
☐	Edit	Copy	Delete	133	2023-01-23 10:21:03	1718	234	245
☐	Edit	Copy	Delete	134	2023-01-23 10:22:01	1718	222	405
☐	Edit	Copy	Delete	135	2023-01-23 10:23:03	1718	234	245
☐	Edit	Copy	Delete	136	2023-01-23 10:23:17	1718	222	180
☐	Edit	Copy	Delete	137	2023-01-23 10:24:01	1725	234	405
☐	Edit	Copy	Delete	138	2023-01-23 10:25:03	1718	222	245
☐	Edit	Copy	Delete	139	2023-01-23 10:26:01	1718	234	405
☐	Edit	Copy	Delete	140	2023-01-23 10:27:03	1718	234	245

Figure 3 : Capture des enregistrements de la BDD

3) Création du site WEB

Le site contient une page « Affichage.php » qui va récupérer les valeurs stockées dans la base de données pour en faire un graphique. Cette page utilise « Highcharts » pour réaliser les graphiques, il est sécurisé en HTTPS et fail2ban pour la connexion.

Dans mon cas, j'ai fait le choix d'afficher que les 30 dernières valeurs et d'avoir la moyenne des mesures affichées.

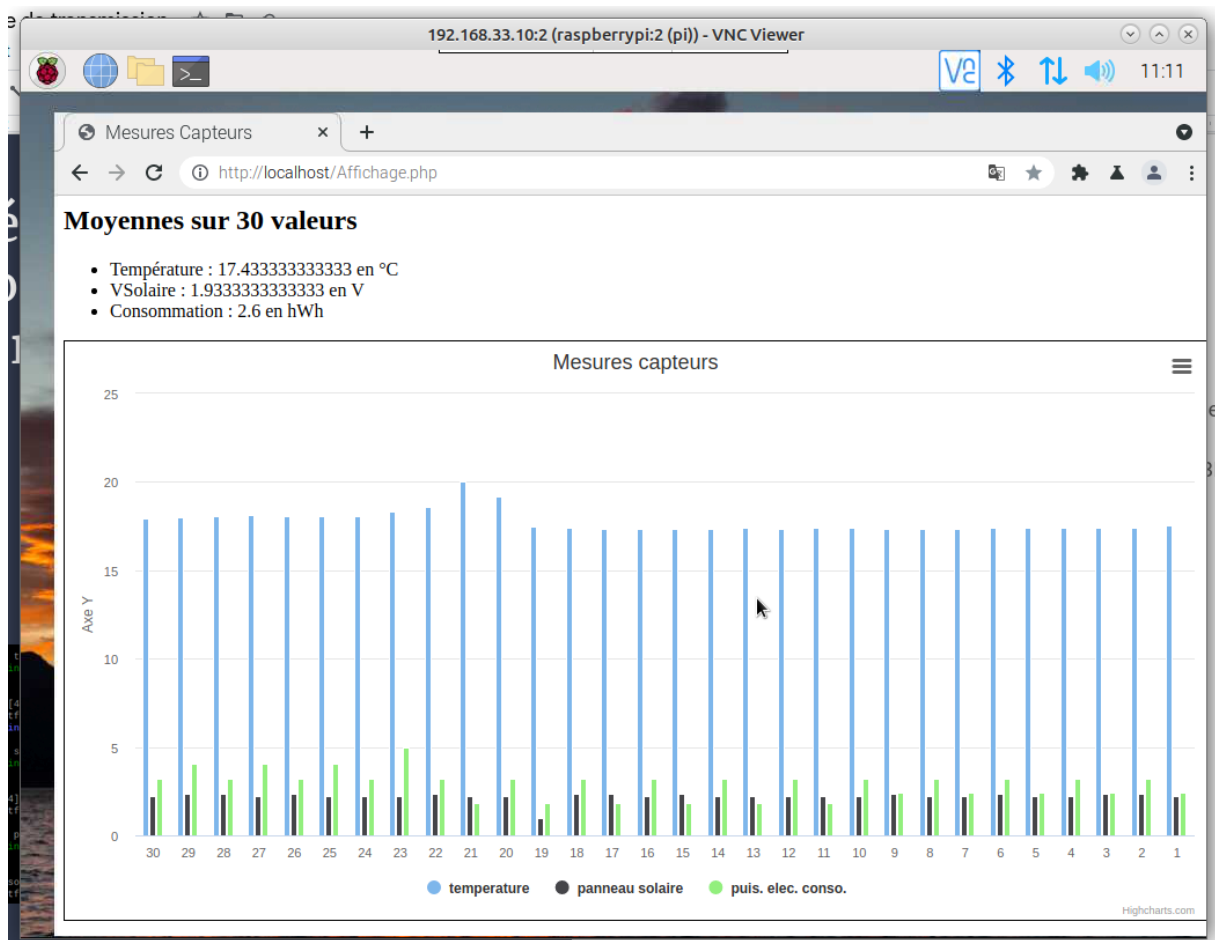


Figure 4 : Capture de "Affichage.php"

The screenshot shows a login form titled 'Affichage.php'. The form has a 'Sign in' header and a URL 'http://localhost'. Below the URL are two input fields: 'Username' and 'Password'. At the bottom right of the form are two buttons: 'Cancel' and 'Sign in'.

Figure 5 : Champs connexion au serveur

4) Récupération des données auprès du serveur

Afin de récupérer les données, j'ai utilisé une clé zigbee vers USB. J'ai lu les messages échangés à l'aide de minicom. Par la suite, j'ai créé un code en C afin de pouvoir me connecter sur la clé zigbee, récupérer les messages du serveur, isoler le message en parties pour récupérer les données, créer une requête SQL pour remplir la BDD et enfin automatiser le tout avec Cron.



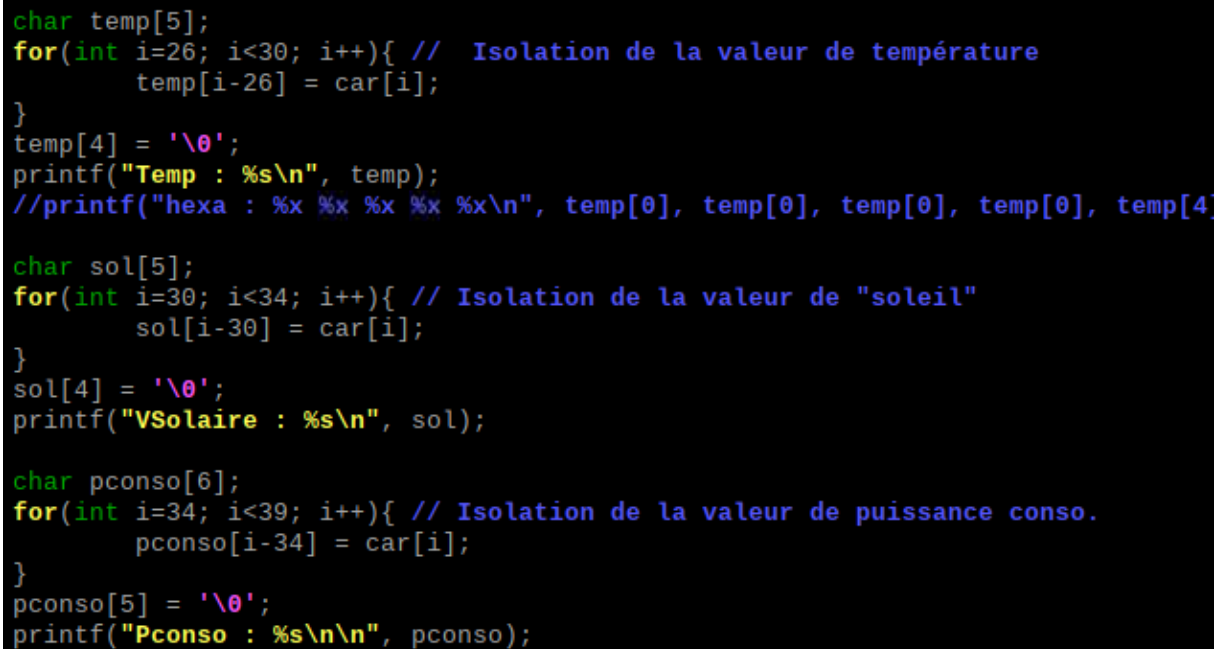
```
Welcome to minicom 2.8

OPTIONS: I18n
Port /dev/ttyUSB0, 10:03:44

Press CTRL-A Z for help on special keys

BCAST:000D6F000C2BC948,0D=1737022204500
BCAST:000D6F000C2BC948,0D=1737023403645
```

Figure 6 : Capture de minicom



```
char temp[5];
for(int i=26; i<30; i++){ // Isolation de la valeur de température
    temp[i-26] = car[i];
}
temp[4] = '\0';
printf("Temp : %s\n", temp);
//printf("hexa : %x %x %x %x %x\n", temp[0], temp[0], temp[0], temp[0], temp[4]);

char sol[5];
for(int i=30; i<34; i++){ // Isolation de la valeur de "soleil"
    sol[i-30] = car[i];
}
sol[4] = '\0';
printf("VSolaire : %s\n", sol);

char pconso[6];
for(int i=34; i<39; i++){ // Isolation de la valeur de puissance conso.
    pconso[i-34] = car[i];
}
pconso[5] = '\0';
printf("Pconso : %s\n\n", pconso);
```

Figure 7 : Code en C

5) Création d'un HotSpot WiFi

Le hotspot wifi est mis en place sur le RPI car ce dernier sert de routeur wifi en utilisant « hostapd ». J'ai mis en place un serveur DHCP afin de fournir des adresses IP aux clients wifi.

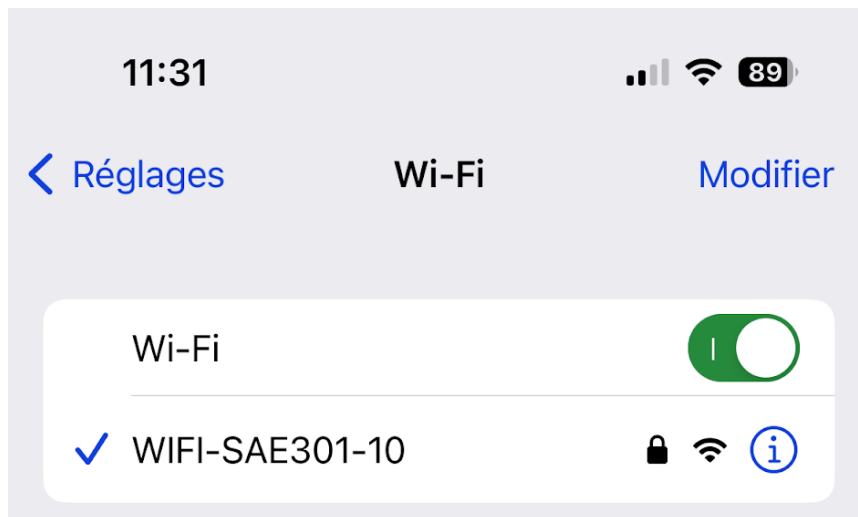


Figure 8 : Capture paramètre wifi sur le téléphone

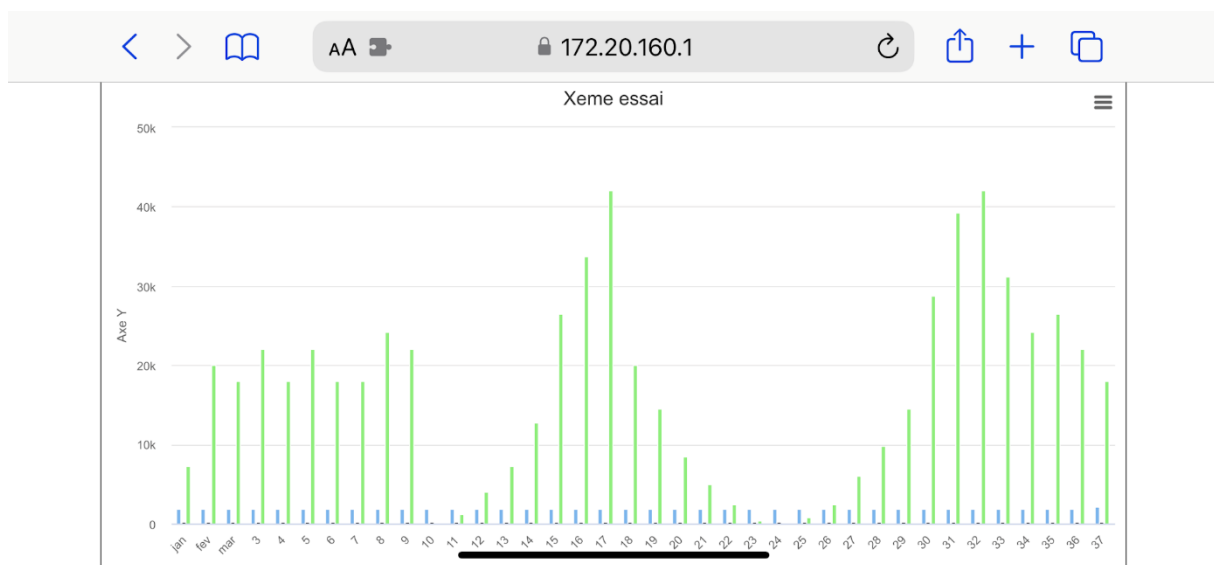


Figure 9 : Capture page web sur le téléphone

6) Mise en place d'un système d'alertes avec SIP

Le serveur SIP est lui aussi hébergé sur le RPI, c'est aussi le client 101. J'ai créé une automatisation qui remplit la condition suivante : si température supérieure à 25 degrés, appel le client 101.

```
if(tempInt >= 25){
    system("/home/pi/script_appel.sh");
}
```

Figure 10 : Capture condition appel

Cette automatisation est une simulation d'appel d'astreinte pour un technicien par exemple.

```
pi@raspberrypi:~ $ cat appel.call
Channel: SIP/100
Callerid: 101
MaxRetries: 2
WaitTime: 20
Context: intrusion
Extension: s
```

Figure 11 : Fichier définissant l'appel d'urgence SIP

```
pi@raspberrypi:~ $ cat script_appel.sh
#!/bin/bash

sudo cp /home/pi/appel.call /var/spool/asterisk/outgoing/
```

Figure 12 : Script permettant de déclencher l'appel