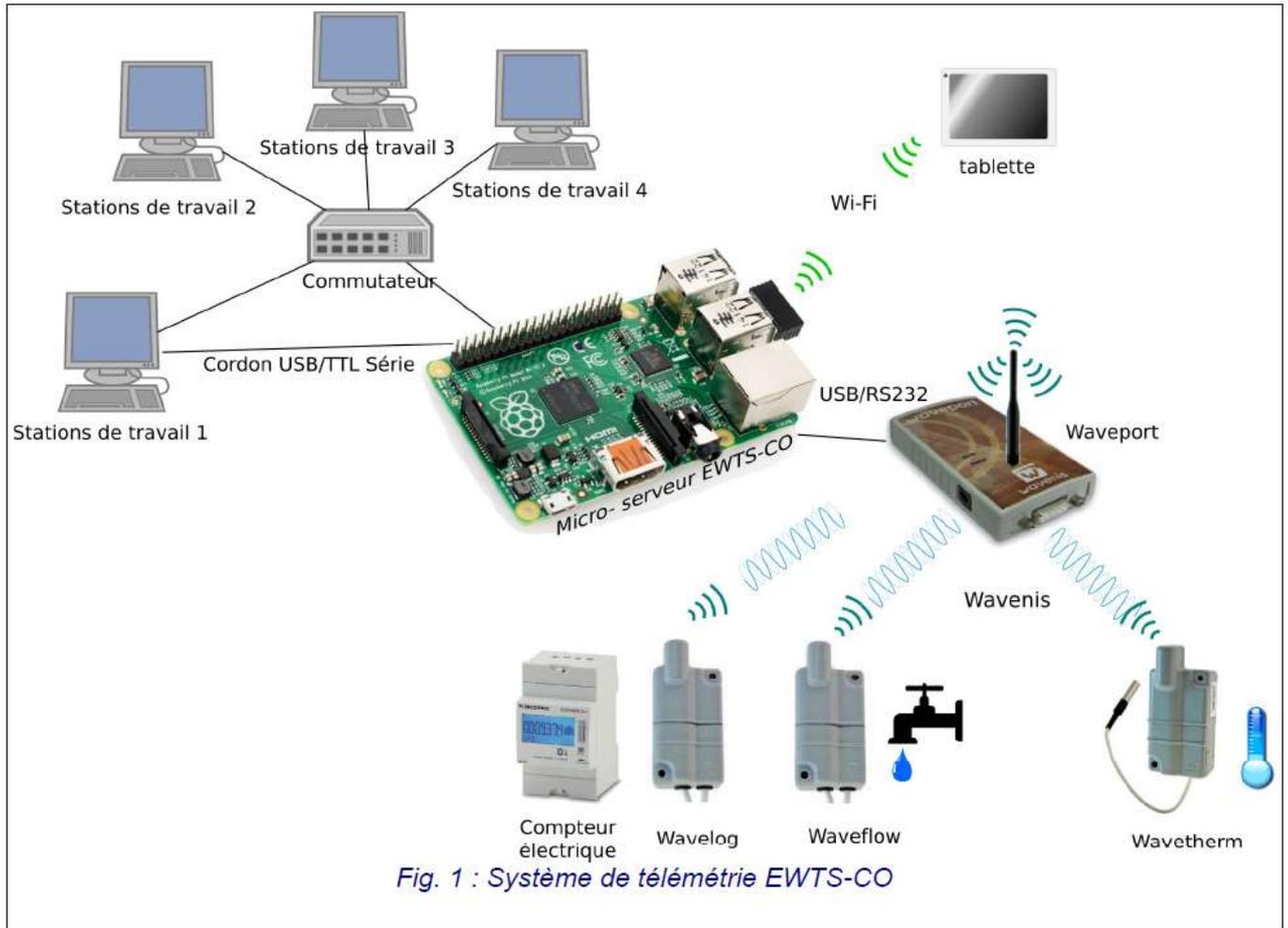


Le micro serveur est fourni avec une carte micro SD avec le logiciel usine, sans configuration. Toutes les informations sont fournies dans le dossier technique joint à ce TP.

1 – Configuration de base

Dans cette première partie, on souhaite réaliser la configuration de base suivante.



Mise sous tension et démarrage :

On souhaite relier le poste de travail avec le cordon USB/TTL au micro serveur EWTS-CO.

⇒ Avant la mise sous du micro serveur, réaliser la liaison et faire vérifier par le prof.

⇒ Vérifier la présence du port série dans le gestionnaire de périphérique sur le poste de travail

⇒ Configurer Putty pour recevoir les informations du micro serveur (mode console).

⇒ Réaliser le câblage de l'installation : le micro serveur a été déclaré sur le serveur du lycée, qui lui affectera une adresse IP en DHCP (sur port Ethernet).

⇒ Mettre sous tension le Waveport et le micro serveur. Observer les informations de démarrage fournies sous Putty.

⇒ Sous Putty, se connecter au serveur.

⇒ A l'aide de la commande ifconfig, noter les adresses MAC et IP pour l'interface Ethernet et pour l'interface WiFi.

⇒ A l'aide du logiciel DipiScan, vérifier la présence du serveur sur le réseau.

Configuration sous le navigateur WEB

Remarque : Le système doit normalement être configuré pour transmettre quelques mesures par jour : autonomie des batteries – interdiction de transmettre plus de 1% du temps dans la bande 868MHz. Mais dans le cadre du test en TP, les mesures se feront toutes les minutes.

Procéder à la configuration du système en suivant la démarche suivante :

- Connexion au serveur à l'aide du navigateur WEB
- Enregistrement des Licences (optionnel comprise)
- Vérification de la date et de l'heure
- Pas d'ajout d'utilisateur supplémentaire (utilisation en mode admin uniquement).
- Création du micro serveur
- Configuration de l'interface Waveport : Sleeping period 60
- Configuration du capteur Wavetherm : Nom = TEMP – Sample period = 60
- Configuration du capteur Waveflow : Nom = DEBIT – Sample period = 60
- Configuration du capteur Wavelog : Nom = CONSO – Sample period = 60
- Vérification de la présence du WavePort sur le port USB0
- Démarrage du "sensorsmanager"

Vérifier le fonctionnement des mesures :

- Consulter les mesures brutes
- Tracer le graphique d'évolution du capteur de température

Dans le menu de configuration avancée des capteurs

⇒ Sélectionner le capteur Wavetherm

⇒ Faire une demande de nouvelle mesure de température et relever la requête et la réponse.

Dans le menu Réseau :

⇒ Déterminer le mode de fonctionnement du WiFi sur le serveur

⇒ A l'aide d'une tablette ou d'un smartphone, se connecter au WiFi du serveur et vérifier l'accès au serveur.

⇒ Indiquer l'adresse attribuée par le service DHCP (en WiFi)

2 – Bilan des communications

Ci-dessous est donné le diagramme de déploiement du système (incomplet car le poste de travail n'apparait pas ici).



La carte interface Wavelog permet d'adapter les impulsions fournies par le compteur au capteur Wavelog.

⇒ Compléter le diagramme de déploiement sur le document réponse en traçant les liaisons.

⇒ A partir de la documentation du Waveport, indiquer les caractéristiques de la liaison RS232 du Waveport.

⇒ A partir de la documentation du module Wavelog (qui donne l'information), indiquer les caractéristiques (fréquence porteuse, vitesse de transmission, type de modulation) de la liaison radio WAVENIS.

3 – Protocole de communication Wavenis

Un résumé du protocole d'échange est donné en annexe.

En validant le mode DEBUG sur la liaison RS232, on a pu récupérer, dans le fichier log.txt, les informations circulant sur cette liaison.

⇒ A l'aide du résumé sur le protocole d'échange, décoder la trame encadrée ci-dessous en complétant le document réponse.

⇒ Montrer qu'il s'agit d'une demande d'envoi de trame (REQ_SEND_FRAME)

```
[15:39:12:964] DEBUG protocol_write_radio_params: Proceeding with writting of w$
[15:39:12:964] DEBUG write_static_standard_params: Entered
[15:39:12:965] DEBUG write_static_standard_params: after write_dynamic_expert_p$
[15:39:12:965] DEBUG write_static_expert_params: Entered
[15:39:12:965] DEBUG write_static_expert_params: after write_dynamic_expert_par$
[15:39:12:966] DEBUG protocol_write_radio_params: write additional command
[15:39:12:966] DEBUG protocol_write_radio_params status 0 OK_PROTOCOL_SUCCESS
[15:39:12:977] DEBUG port_send: Enter function
[15:39:12:977] DEBUG port_send [6] port_send START sending Status:0 and status:1
[15:39:12:977] DEBUG port_send RS232Driver send: FF020B2010194C3007A401C46203
[15:39:12:996] DEBUG process_port: /dev/ttyUSB0, -----$
[15:39:12:997] DEBUG process_port: /dev/ttyUSB0, amountInInQueue = 0
[15:39:12:998] DEBUG process_port: STATUS_SEND_COMMAND
[15:39:12:998] DEBUG process_port: /dev/ttyUSB0 - STATUS_SEND_COMMAND: 0
[15:39:12:999] DEBUG porcess_port: End states processing with processedOk=0, er$
[15:39:13:040] DEBUG process_port: /dev/ttyUSB0, -----$
[15:39:13:040] DEBUG process_port: /dev/ttyUSB0, amountInInQueue = 15
[15:39:13:040] DEBUG process port: STATUS WAIT ACKNOWLEDGE
```

⇒ Préciser à quel capteur est destinée cette demande

⇒ A l'aide de la documentation du capteur concernée, déterminer le type demande réalisée (voir APPLICATIVE COMMANDS)

La réponse à la requête précédente a également été récupérée dans le fichier log.txt et elle est donnée ci-dessous:

```
DEBUG RS232Driver IDLE recv: FF02113010194C3007A4810A8001484FFF6E2603
```

⇒ De la même manière, décoder la trame sur le document réponse.

⇒ Montrer qu'il s'agit d'une réponse du capteur Wavetherm et déterminer la valeur de la température envoyée.