

La station Météo de l'ASAP Lille

Le 12 septembre 2023

La station météo est composée d'un module central et de plusieurs capteurs. Actuellement il existe 6 types de capteurs qui peuvent être connectés à la station météo. Cette station propose un point d'accès WIFI qui pourra être utilisé pour suivre les remontées d'information des capteurs dans une page web. Il peut optionnellement être connecté à internet pour initialiser la date et l'heure du jour de sa mise sous tension. Dans le cas où la station n'est pas connectée à internet, la date et l'heure courante sera initialisée en chargeant la page web à cette adresse IP : 192.168.4.1

L'aspect matériel

Le module central de la station météo est composé d'une carte ESP32-CAM qui est enfichée dans une autre carte qui est utilisée pour la connexion USB (POPESQ® 1 pcs. x ESP32-CAM-MB Programmeur #A4444) pour y injecter le programme et pour l'alimenter. A noter que la fonction CAM de la carte n'est pas utile mais il a été choisi car il embarque une carte SD.

Ce module central est lui-même organisé en plusieurs modules logiciels ayant chacun leur objectif :

- Module CONFIG : il assure la configuration et le paramétrage de l'application
- Module ESP-NOW : il assure collecte des données provenant des capteurs
- Module STOCKAGE : il assure le stockage des mesures sur la carte SD

Il faudra prévoir un chargeur de téléphone avec prise USB pour alimenter le module central de la station météo.

Le module CONFIG

Ce module regroupe un ensemble de fonctions et de variables qui vont donner de la souplesse au code en extériorisant les paramètres de fonctionnement.

a) Les paramètres du WIFI

Ils sont définis dans les 4 premières lignes du fichier « **config.txt** » qui est préparé pour un fonctionnement sans connexion à internet :

1. Nom du réseau de la station : valeur initiale dans le fichier **stationMeteo**
2. **SSID** de la box WIFI du domicile : ligne vide si pas de connexion internet ou ssid de la box
3. **Mot de passe** de la box WIFI : ligne vide si pas de connexion internet
4. **Nom du serveur NTP** utilisé pour récupérer la date et l'heure courante : ligne vide si pas de connexion internet ou **pool.ntp.org**

b) Les paramètres des capteurs

Ils sont définis dans le fichier **boards.csv** au format CSV (dont le caractère ; est le séparateur de champ) et est modifiable soit par un éditeur de texte (comme Notepad++) soit par calc ou excel. Les différents d'une ligne sont :

- id : l'identifiant du capteur (une valeur numérique de 1 à ...)
- owner : le nom du responsable de ce capteur
- typemesures : une chaîne de caractères composée d'une suite d'un à trois types de mesure (typemesure) séparé par le caractère |. Chaque typemesure est identifié par un seul caractère.
- Unites : une chaîne de caractères composée d'une à trois unités (unite) séparées

par le caractère |. Chaque unité est elle-même une chaîne de caractères qui ne doit pas contenir ni le caractère | ni le caractère ;

- titles : une chaîne de caractères composée d'un à trois « titles » (title) séparés par le caractère |. Chaque unité est elle-même une chaîne de caractères qui ne doit pas contenir ni le caractère | ni le caractère ;

Voici la liste des caractères qui ont été choisis arbitrairement pour définir chacun un type de mesure :

- a : vitesse du vent
- b : pression
- g : sens du vent
- h : hygrométrie
- n : niveau d'eau de pluie
- t : température

Par exemple, la carte du Pluviomètre gérée par Claude qui envoie une seule mesure de niveau d'eau recueillie par le capteur est représentée par cette ligne :

3;Claude;n;mm;Pluviometre

Pour la carte gérée par philippe qui envoie la température, la pression et l'hygrométrie, la ligne est :

1;Philippe;t|p|h;C|b|%;temperature|pression|hygro

c) Affichage d'une mesure

L'affichage d'une mesure est contrôlé par le fichier **board.html**. Le fichier comprend un fragment de code html dans lequel apparaissent les variables %id%, %owner%, %typemesure%, %unite% et %title% propres à chaque type de mesure. Bien entendu, entre les caractères % nous retrouvons le nom de chaque variable définie dans le fichier **board.csv**.

d) affichage de la page «tableau de bord »

La page HTML qui affiche le tableau bord de la station météo est contenue dans le fichier « **index.html** » lui aussi au format HTML. Lorsqu'il est envoyé au navigateur, l'application remplace la variable %boards% calculée à partir d'une agrégation de toutes les mesures.

Fonctions de ce module :

- lancement de l'initialisation générale → setup()
- lancement de la configuration générale de tous les modules → config()
- affichage d'un fichier (utile au debug) → printFile()
- Initialisation d'une variable de type HTLM à partir du contenu d'un fichier → readHtml()
- Initialisation des capteurs → initBoards()
- Initialisation du numéro de la prochaine mesure lors du démarrage → initNextMeasureId()
- Teste l'état de fonctionnement du WIFI → isWiFiConnected()
- Création de la liste des mesures à stocker → creerListeMesures()
- Retrouve le « title » d'un capteur → getBoardValue()
- Retrouve la description des types de mesure d'un capteur → getBoardString()
- Retrouve toute la chaîne qui décrit un capteur → getValue()
- Retrouve le ième type de mesure d'un capteur → getBoardValueAsChar()

- génère sous forme de chaîne de caractères la description complète d'un capteur (debug) → boardToString()
- retourne la structure de données du capteur à partir de son identifiant → getBoardById()
- retourne le nombre de types de mesure supportés par un capteur → getNumberOfValues()
- retourne la date et heure [au démarrage] depuis le serveur NTP → getTime()
- retourne la date et heure courante → getCurrentTime()
- retourne sous forme lisible par un humain la date et l'heure → timeToString()
- affiche l'adresse MAC du module central → printMAC()
-

Le module ESP-NOW

Le module ESP-NOW collecte tous les messages envoyés par chacun des capteurs.

Les capteurs envoient 2 types de message :

- un message de demande d'appairage sur le canal WIFI utilisé par le module central (type de message = 0)
- un message contenant des mesures (type de message = 1)

Le module central envoie un seul de type de message pour répondre positivement à la demande d'appairage

La structure de chacun de ces messages est ainsi définie :

Message de demande d'appairage sur le canal WIFI :

```
typedef struct struct_pairing {
    uint8_t msgType;
    uint8_t id;
    uint8_t macAddr[6];
    uint8_t channel; // a été rendue no obligatoire !
} struct_pairing;
```

Pour répondre positivement à une demande d'appairage, le module central envoie la même structure de message avec msgType=1 et indique son adresse mac et le numéro de canal qu'il utilise.

Voici la structure du message envoyé par les capteurs pour envoyer des mesures :

```
// taille maximale = 250
// ici taille = 18 octets
typedef struct struct_message {
    uint8_t msgType; // 1 octet
    uint8_t id; // 1 octet
    unsigned int seqNum; // 4 octets → n'est pas exploitée dans cette version (mais
elle est stockée sur la carte SD)
    float valeurs[10]; // taille = 10 * 4 = 40 octets -> on pourrait avoir float valeurs[60]
} struct_message;
```

Une fois un message de mesures reçu, l'application, crée la ou les mesures. Une mesure est créée pour chaque valeur envoyée.

Ensuite chaque mesure est stockée sur la carte SD et la page est mise à jour par exécution d'un code javascript déclenché par le module central.

Le module central peut actuellement gérer jusqu'à 10 mesures par capteur, mais les tests ont jusqu'ici été limité à 3 mesures.

Fonctions de ce module :

- Initialise le module → `initESP_NOW()`
- établit l'appairage WIFI avec un capteur → `addPeer()`
- création d'une mesure à partir d'une message reçu → `creerMesure()`
- renvoie dans une chaîne de caractères la mesure pour affichage (debug) → `mesureToString()`
- renvoie le type de mesure → `getTypeld()`
- renvoie la valeur d'une mesure → `getValue()`
- renvoie l'horodatage d'une mesure → `getTimeStamp()`
- renvoie la carte qui a généré la mesure → `getBoard()`
- renvoie l'identifiant de la mesure → `getMesureId()`
- Réception d'un message → `OnDataRecv()`
- Après l'envoi d'un message, en affiche le statut → `OnDataSent()`
- ajoute une nouvelle mesure à la liste des mesures à stocker → `ajouterMesure()`

Le module STOCKAGE

Ce module remplit séquentiellement le fichier **mesures.csv** au format CSV (séparateur = le caractère;). Chaque mesure est représentée par une ligne. Une ligne contient une chaîne de caractères contenant plusieurs champs :

- numéro de la mesure (numérotation séquentielle)
- id du capteur
- le numéro de séquence de la mesure (seqnum) envoyé par le capteur
- la date et heure de la réception de la mesure
- la valeur de la mesure

Fonctions de ce module :

- initialisation du gestionnaire de la carte SD → `initSDCard()`
- test de bon fonctionnement de la carte SD → `testSDCard()`
- écriture d'une mesure dans le fichier mesures.csv → `appendFile()`
- stockage d'une mesure sur la carte et gestion de la liste des mesures → `storeMesure()`
- renvoie dans une chaîne de caractères les mesures comprises entre deux bornes (from et to) → `readMesures()`
- renvoie le nombre de mesures stockées sur la carte → `readNbMesures()`
- renvoie au format csv la description d'une mesure → `measureToCSV()`

La fonction principale

- boucle principale d'exécution → `loop()`