

# Rapport de laboratoire

---

**École supérieure**  
Électronique

Laboratoire PROG  
Salle R110

---

## Capteur Météo Wifi

---

**Réalisé par :**

Vullioud

**À l'attention de :**

M. Bovey



---

**Table des matières :**

Capteur Météo Wifi.....	1
1 Cahier des charges du projet .....	5
2 Travail à effectuer .....	5
3 Mise en service.....	6
3.1 Création du réseau .....	6
3.2 Installation du serveur Domoticz .....	6
3.3 Mise en marche du prototype.....	7
4 Modification du programme .....	9
4.1 Test du projet sous MPLAB v5.45 et Harmony v2.06 .....	9
4.2 Modification de la structure Wifi.....	9
4.3 Affichage de l'IP du périphérique sous Info réseau .....	9
4.4 Correction des dérivées de la pression.....	10
4.5 Ajout de la prise en compte de l'altitude .....	10
5 État du projet.....	11
6 Amélioration .....	11
7 Conclusion .....	11
8 Annexes.....	12
8.1 Rapport de projet initial .....	12
8.2 Listing du code.....	12



# 1 Cahier des charges du projet

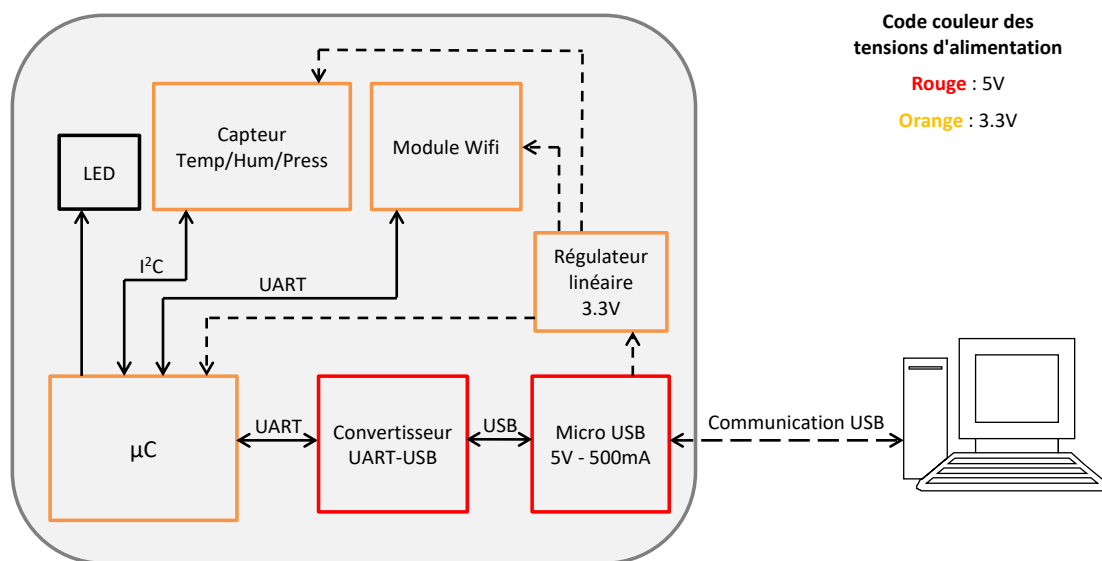
Le but de ce projet sera de réaliser un capteur météorologique permettant de communiquer ses mesures et prévisions via Wifi sur Domoticz.

Domoticz est un système domotique qui vous permet de surveiller et de configurer divers appareils tels que: lumières, commutateurs divers capteurs/compteurs tels que la température, la pluie, le vent, les UV, l'électricité, le gaz, l'eau et bien plus encore.

Pour se faire, sur la carte se trouvera un capteur de température, humidité et pression sur lequel nous viendrons récupérer les données toutes les 20min. Grâce aux données de pression, nous pourrons créer un algorithme de prévision de la météo.

Les données et les prévisions seront envoyées via des requêtes HTTP à travers un module Wifi ESP click.

Un connecteur USB sera aussi présent afin d'alimenter la carte, mais aussi permettant de communiquer avec le microcontrôleur depuis un ordinateur via un convertisseur UART-USB.



Nous pouvons voir ci-dessus le schéma bloc de la carte.  
Se référer à l'annexe, pour plus de détail.

## 2 Travail à effectuer

- Mettre en place un réseau wifi et un serveur Domoticz
- Remettre en service le capteur
- Porter le projet MPLABX vers la version actuellement utilisée
- Nettoyer l'algo de prévision météo ( fonction BME280\_Forecast() )
- Faire en sorte que l'altitude réglée soit prise en compte
- Utiliser la LED RGB pour indication de l'état
- Optimiser le programme pour une économie d'énergie

## 3 Mise en service

### 3.1 Création du réseau

Afin de pouvoir mettre en service notre prototype, nous allons devoir configurer un réseau wifi sur un routeur que nous avons emprunté à M.Castoldi.

Le nom du SSID sera Wifi-Projet-2002 et le mot de passe sera 2002-CapteurMeteo.

**WIRELESS NETWORK SETTINGS**

Enable Wireless : ☒ Always

Wireless Network Name :  (Also called the SSID)

Enable Auto Channel Selection : ☒

Wireless Channel :

Transmission Rate :  (Mbit/s)

Wireless Mode :

Band Width :

Enable Hidden Wireless : ☐ (Also called the SSID Broadcast)

**WIRELESS SECURITY MODE**

Security Mode :

**WPA/WPA2**

WPA/WPA2 requires stations to use high grade encryption and authentication.

Cipher Type :

PSK / EAP :

Network Key :  (8~63 ASCII or 64 HEX)

Nous allons utiliser une machine virtuelle Windows 10 qui sera connectée sur notre routeur. Nous aurons ainsi un réseau composé de notre prototype connecté en wifi et notre machine virtuelle connectée au routeur.

### 3.2 Installation du serveur Domoticz

Nous allons installer notre serveur Domoticz sur notre machine virtuelle Windows 10.

**Domoticz**

Accueil Interrupteurs Scénarios Température Météo Mesures Configuration

Afficher 25 entrées Recherche :

Idx	Nom	Activé	Type	Adresse	Port	Délai d'inactivité
3	2002-CapteurMeteoWifi	Oui	Dummy (Does nothing, use for virtual switches only)			Désactivé

Affichage de 1 à 1 sur 1 entrée(s) Première Précédente 1 Suivante Dernière

Modifier Supprimer

Actif: ☒

Nom:

Type:

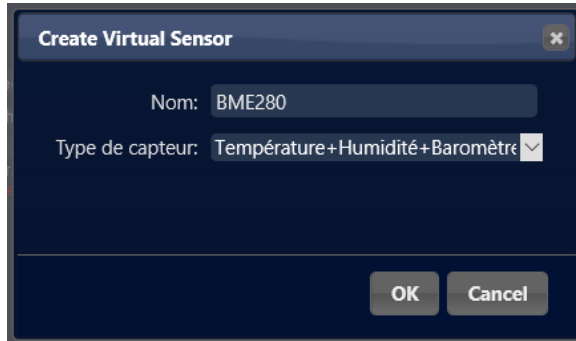
Délai d'inactivité:

Le périphérique sera redémarré si aucune donnée n'est reçue pendant ce délai.  
N'activez pas cette option pour les périphériques ne recevant pas de données !

Ajouter

Nous avons pu installer notre serveur. Nous pouvons nous connecter à celui-ci en tapant 127.0.0.1 :8080 sur notre moteur de recherche. Nous allons ajouter un périphérique en le nommant ESP8266 HTTP.

Une fois le périphérique créé nous allons lui rajouter un capteur, qui comprendra la température, l'humidité et la pression. Nous pourrons le faire en appuyant sur créer un capteur virtuel.



En appuyant dessus nous ouvrons la fenêtre ci-dessus. Nous allons nommer notre capteur BME280.

Une fois les étapes précédentes effectuées nous obtenons la fenêtre ci-dessous.



	Idx	Matériel	ID	Unit	Nom	Type	Sous-type	Données	Statut	Dernier contact
<input type="checkbox"/>	1	2002-CapteurMeteoWifi	14051	1	BME280	Temp + Humidity + Baro	THB1 - BTHR918, BTHGN129	0.0 C, 50 %, 1010 hPa		2020-08-31 10:12:58

Nous pouvons voir que nous n'avons pas encore le transfert de données activé, elles ne sont pas à jour.

### 3.3 Mise en marche du prototype

Nous pouvons maintenant allumer notre prototype, puis régler l'IP du serveur Domoticz afin que celle-ci soit celle de l'IP de la machine virtuelle.



	Idx	Matériel	ID	Unit	Nom	Type	Sous-type	Données	Statut	Dernier contact
<input type="checkbox"/>	1	2002-CapteurMeteoWifi	14051	1	BME280	Temp + Humidity + Baro	THB1 - BTHR918, BTHGN129	27.1 C, 31 %, 1024 hPa		2020-08-31 10:24:16

Nous avons maintenant notre transfert de données qui est actualisé. Les mesures semblent correctes, nous pouvons dire que notre périphérique est mis en fonction.





## 4 Modification du programme

### 4.1 Test du projet sous MPLAB v5.45 et Harmony v2.06

Nous pouvons maintenant venir prendre en main le programme. Nous allons copier le programme initial afin de le mettre dans le dossier « apps » de Harmony.

Nous pouvons ouvrir le projet puis compiler celui-ci sans problème. Une fois notre carte programmée notre prototype fonctionne correctement.

### 4.2 Modification de la structure Wifi

J'ai pu remarquer que la variable de la structure représentant l'IP du serveur Domoticz n'était pas claire j'ai pu changer son nom afin que celle-ci soit parlant.

Nous avons l'endroit où nous nous connectons au serveur, nous allons alors communiquer l'IP de destination ainsi que le port. Celui-ci sera alors la variable IPDom de la structure Wifi.

```
if(ESP8266_ConnectToTCPServer(Wifi.IPDom,Wifi.PortDom) == OK)
{
    SendWifiMessage(cmdString, cmdStringLength);
    delay_msCt(10);
    SendWifiMessage(httpString, httpStringLength);
}
```

Nous avons alors la structure suivante.

```
//Structure du Wifi
typedef struct Wifi {
    char SSID[100];    // Nom du réseau
    char PWD[100];     // Mot de passe du réseau
    char IPDom[15];    // Adresse IP du serveur Domoticz
    char PortDom[4];   // Port du serveur Domoticz
```

### 4.3 Affichage de l'IP du périphérique sous Info réseau

Une des choses que j'ai pu remarquer c'est que nous n'affichions pas l'IP réel de notre machine lors de l'affichage des informations du wifi.

Nous allons donc regarder la où nous prenons notre IP.

```
//si oui, test de le connecter au Wifi
if(ESP8266_ConnectWifi(Wifi.SSID,Wifi.PWD) == OK)
{
    //si oui, le met en mode station et on enregistre son adresse IP et MAC
    ESP8266_Mode(ModeStation);
    ESP8266_IP(ESP8266.IP, ESP8266.MAC);
```

Nous allons afficher notre IP réel, celle qui correspond à notre périphérique.

```
sprintf((char*)Line1, "SSID : %s\r\n", Wifi.SSID);
sprintf((char*)Line2, "PWD  : %s\r\n", Wifi.PWD);
sprintf((char*)Line3, "IP   : %s\r\n", ESP8266.IP);

SendUSBMessage(Line1, strlen((char*)Line1));
SendUSBMessage(Line2, strlen((char*)Line2));
SendUSBMessage(Line3, strlen((char*)Line3));
```

## 4.4 Correction des dérivées de la pression

Dans le précédent programme, nous effectuions la dérivée de chacune des mesures par rapport à la mesure actuelle puis nous faisons la moyenne, pour avoir la différence de pression moyenne en 3h. Cette façon de faire n'est pas correcte, il faut faire la dérivée entre chaque point puis en faire la moyenne.

Nous avons ci-dessous la version corrigée de la moyenne de la dérivée sous 3h.

```
//moyenne de toutes les dérivées entre les mesures
for(cMoyenne = 0; cMoyenne <= 8; cMoyenne++)
{
    PressionMoy += (Pression[cMoyenne]- Pression[cMoyenne+1])*3;
}

PressionMoy /= 9;
```

## 4.5 Ajout de la prise en compte de l'altitude

Dans le cahier des charges, il nous est demandé de prendre en compte l'altitude afin de prévoir la météo. Cela peut se faire, car dans notre fonction de prévision météo il y a un moment où nous comparons la pression mesurée avec une valeur normalisée de pression, si elle est plus grande alors il fait plutôt beau alors que si elle est en dessous il fait plutôt mauvais temps.

Grâce à l'altitude, j'ai pu déterminer la pression que nous devrions retrouver normalement. Comme nous pouvons le voir ci-dessous.

```
//test si la dernière mesure de la pression est plus grande que
//La pression normalisée en fonction de l'altitude
if(BME280.Pression >= (1013.25*exp(-(BME280.Altitude/8400))))
{
    BME280.Pression_Forecast = 1;    //Prévision : beau temps (Sunny)
}
else
{
    BME280.Pression_Forecast = 2;    //Prévision : nuageux (Cloudy)
}
```

## 5 État du projet

Nous avons un prototype fonctionnel, il est mis en service avec la dernière version du programme. Tout est fonctionnel cependant il reste certain bug dans le menu.

## 6 Amélioration

Nous n'avons pas pu faire que la LED indique l'état de la météo. Nous n'avons pas fait d'optimisation de l'énergie, les mesures sont censées se faire chaque 20min alors nous pouvons surement le mettre dans un mode d'économie entre temps.

Nous avons pu voir une série de bugs sur le menu qui rendent difficile l'utilisation de celui-ci lorsque les bugs surviennent. Les structures sont elles assez mal imaginées, il serait intéressant de faire du tri afin que l'on puisse facilement travailler avec.

## 7 Conclusion

Le projet est à un point assez avancé, lorsque je l'ai pris en main il était déjà fonctionnel, mais il y avait des choses à modifier. J'ai pu amélioré la grande partie cependant je n'ai pas eu le temps de faire plus, car nous n'avons pas énormément de temps dédié au projet.

Lausanne, le 06 juin 2022

Julien Vullioud

## 8 Annexes

### 8.1 Rapport de projet initial

### 8.2 Listing du code