

## Séance 5: Compte rendu du 3 février

### Bilan

En début de séance, nous avons fait le point sur l'avancé du projet aillant déjà réalisé l'objectif de la séance d'aujourd'hui.

En effet, la mise en commun des programmes a déjà été faite au fur et à mesure donc nous avons anticipé tout les problèmes liées aux entrées/sorties déjà utilisés pour certains capteurs, les librairies etc...

De plus, le programme qui compare les données pour indiquer les seuils de dangerosité sera fait avec l'application et nous avons déjà toutes les informations nécessaire.

Voici un exemple de seuils pour le capteurs de la qualité de l'air :

Un taux en dessous de 50 : Air correct (Fresh air)  
Un taux au dessus de 50 : Qualité médiocre (Low pollution)  
Un taux au dessus de 150 : Air pollué (High pollution)  
Un taux au dessus de 400 : Air très pollué (High pollution alert)

Nous décidons alors aujourd'hui de commencer le grand projet de la semaine prochaine c'est à dire la connexion entre la carte UNO et l'application Blynk.

Je m'occupe de l'application Blynk car nous n'avons pas encore commencé à l'utiliser et on en aura besoin pour la connexion.

### L'application Blynk :

Après avoir effectué des recherches sur cette application. J'ai créé un compte et un nouveau projet puis j'ai recherché ce qu'on pouvait faire sur l'application.

Ce qui est pratique c'est qu'il est décrit pour chaque widgets (un peu comme l'application que nous avons utilisé « Bluetooth Electronics ») comment est ce qu'on peut les programmer par exemple « `Blynk.virtualWrite(V1,val)` ; » pour visualiser la valeur lu par un capteur.

En lisant quelques forums je remarque qu'il faut recevoir un mail de l'application nous précisant un jeton d'authentification, or je ne l'ai pas reçu directement car pour le recevoir il faut tester et se connecter à la carte.

Pour la partie code, il faut de nouveau ajouter une nouvelle bibliothèque. De plus, il faut remplacer `auth` dans « `Blynk.begin(auth, ssid, pass)`; » par le jeton pour permettre à l'application d'associer l'affichage de notre projet (sur le téléphone) et la carte UNO.

## Voici le code sur Arduino

```
#include <SoftwareSerial.h>
#include <DHT.h>
#define DHTPIN A0 //connecté à A0
#define DHTTYPE DHT11 // DHT 11
#define AIR_PIN A1 //connecté à A1
#define BLYNK_PRINT Serial
char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";
DHT dht(DHTPIN, DHTTYPE);
int DUST_PIN = 8; //connecté à I/O 8
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 30000; // 30s
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

// fonction qui récupère et affiche les données du capteur Temp&Humi
void sendSensorTH() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    Serial.print("La température est: ");
    Serial.print(t);
    Serial.println("°C");
    Serial.print("L'humidité est: ");
    Serial.print(h);
    Serial.println("%");
    if (isnan(h) || isnan(t)) { //on check si on a bien des nombres (
        Serial.println("ERROR");
    }
    if (h<40) {
        Serial.println("Humidité basse, attention!");
    }

    if (h>50) {
        Serial.println("Humidité haute, attention!");
    }
}

// fonction qui récupère et affiche les données du capteur AirQuality
void sendSensorAQ() {
    int value = analogRead(AIR_PIN);
    Serial.print("L'indice de pollution est: ");
    Serial.println(value);
    if (isnan(value)) { //on check si on a bien des nombres (
        Serial.println("ERROR");
    }
    if (value>150) {
        Serial.println("Air pollué, attention!");
    }
    if (value>350) {
        Serial.println("Air très pollué, attention danger possible!");
    }
}

// fonction qui récupère et affiche les données du capteur Dust
void sendSensorD() {
    duration = pulseIn(DUST_PIN, LOW);
    lowpulseoccupancy = lowpulseoccupancy+duration;
    if ((millis()-starttime) > sampletime_ms) //temps d'échantillonnage == 30s
    {
        ratio = lowpulseoccupancy/(sampletime_ms*10.0); //pourcentage
        concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; // using spec sheet curve
        Serial.print("La concentration de l'air est ");
        Serial.print(concentration);
        Serial.println("pcs/0.01 cf");
        lowpulseoccupancy = 0;
        starttime = millis(); //restarting
    }
}

void setup() {
    Serial.begin(9600);
    dht.begin();
    pinMode(AIR_PIN, INPUT);
    pinMode(DUST_PIN, INPUT);
    starttime = millis(); //curent time
}

void loop() { //permet de lancer toutes les fonctions toutes les 5 secondes
    sendSensorTH();
    sendSensorAQ();
    sendSensorD(); //celui ci est toutes les 30s
    Blynk.run();
    delay(5000);
}
```

Avec Blynk je peux associer chaque sorties du module wifi à un widget et indiquer comment l'afficher sur l'application, tout les combien je souhaite qu'il s'affiche et avec combien de chiffre significatif.

Pour afficher les données des capteurs il faudra utiliser le widget « Labeled Value» et pour choisir le nombre de décimale on peut utiliser /pin.#/ (une décimale) et /pin.##/ (deux décimales).

Programme l'affichage des données du capteur de poussière sur Blynk :

