

# L'Andain'AIR CUBE :

## Un capteur de dioxyde de carbone en open source

Programme 3 : Pilotage du programme, via Bluetooth, sur une application pour Android ; affichage des valeurs sur une tablette ou smartphone (et export des données au format csv) ; affichage visuel des seuils avec des diodes et des valeurs sur le port série

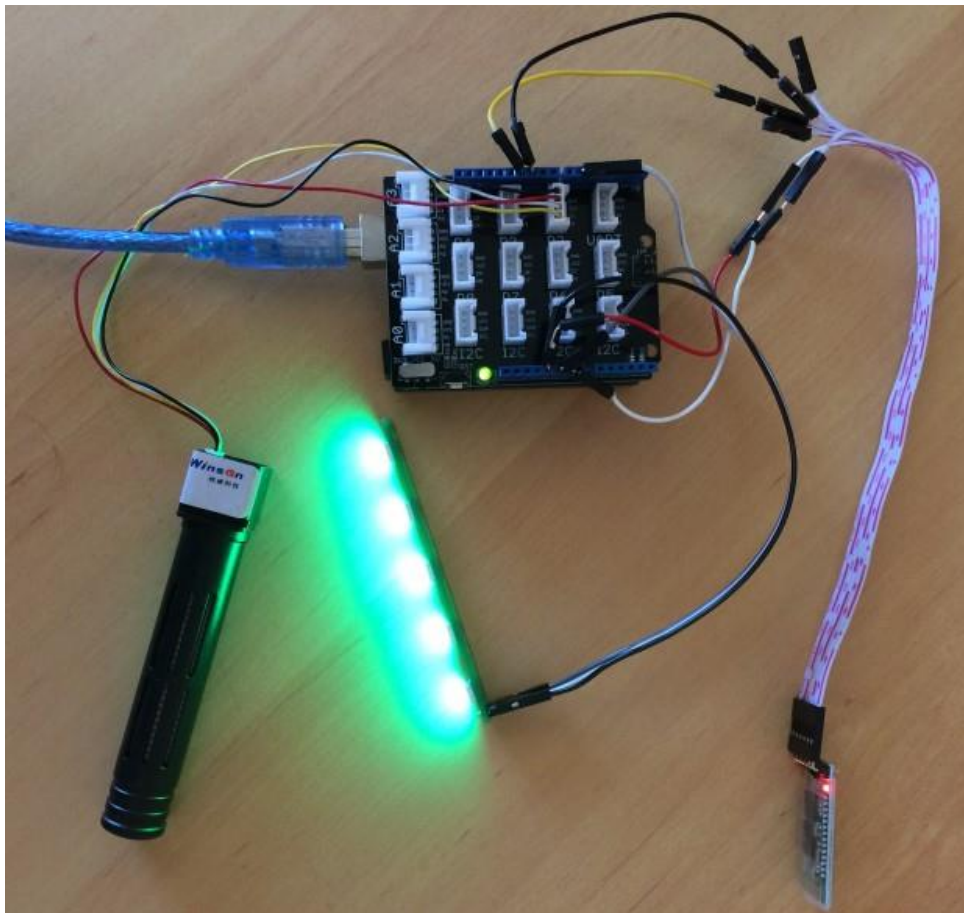
- **Matériel :**

- ✓ Arduino
- ✓ Module grove base shield
- ✓ Capteur de CO<sub>2</sub> grove (Infrared CO2 sensor MH-Z16)  
Exemple : [https://wiki.seeedstudio.com/Grove-CO2\\_Sensor/](https://wiki.seeedstudio.com/Grove-CO2_Sensor/)
- ✓ Un stick de 5 LED NeoPixel
- ✓ Module Bluetooth HC05
- ✓ Une tablette ou smartphone (Android)

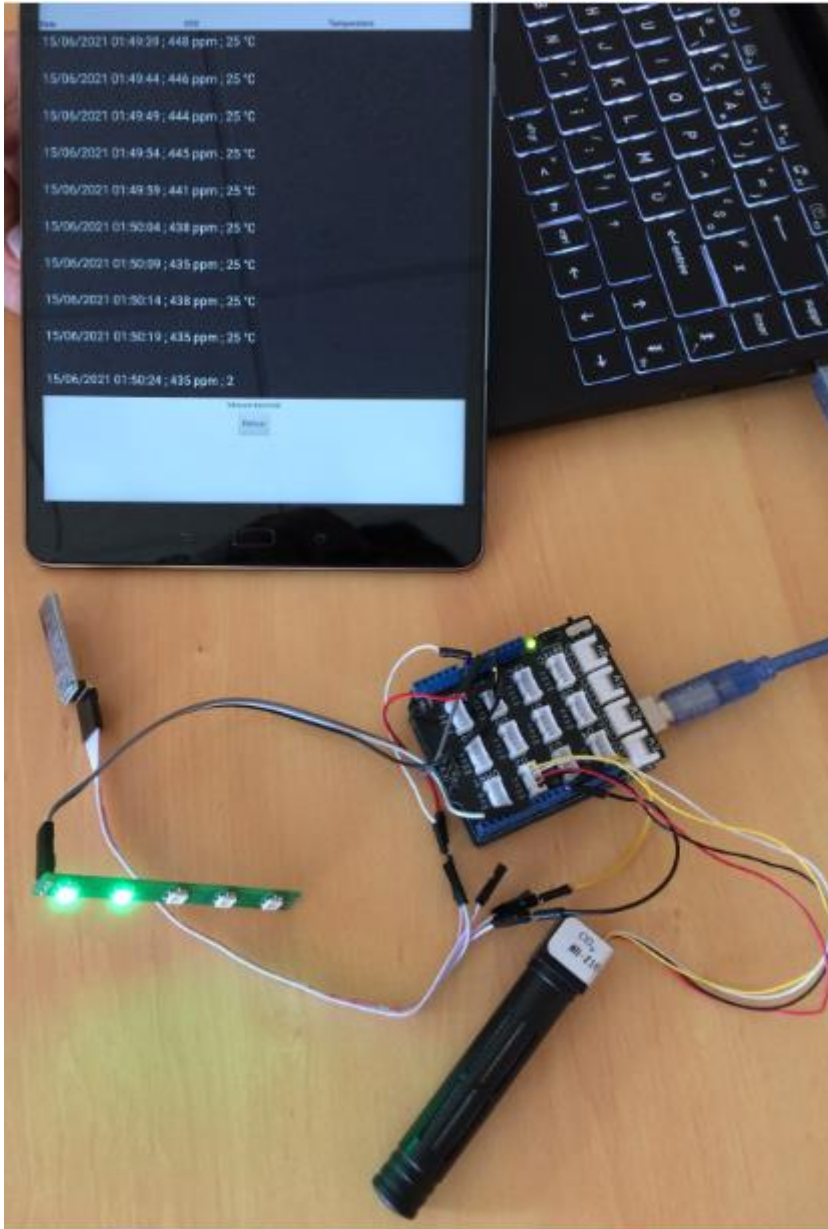


- **Branchements :**

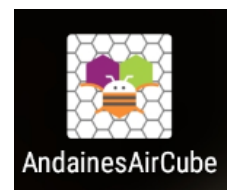
- ✓ Le capteur grove CO<sub>2</sub> se branche en D2 sur le module grove base shield d'Arduino (cf ci-dessous).
- ✓ Stick de LED : 5V à 5V (fil noir sur la photo) ; G à la masse GND (fil gris sur la photo) et D à D~6 (fil blanc sur la photo) (cf ci-dessous).
- ✓ Module Bluetooth HC05 : suivre les indications écrites sur le capteur.



- **Programme :** voir dossier zippé nommé « Programme 3\_pilotage bluetooth\_affichage LED et port série\_fiche explicative ».



- ✓ Installer l'application (fichier AndainesAirCube.apk) sur le smartphone ou tablette (Android).
- ✓ Appairer en Bluetooth le capteur HC05 (avec le code 1234 ou 0000). Cette opération est à faire la première fois seulement.
- ✓ À faire à chaque fois, appuyer sur « connexion » et choisir HC05 pour que « connexion statut » devienne « connexion connecté »



Connexion    Connecté  
Paramètres



- ✓ Appuyer sur « Paramètres »

Paramètres

Durée (en s) 600

Intervalle (en s) 50

Seuil 1 : 2 LED vertes 800

Seuil 2 : 2 LED orange 1000

Seuil 3 : 3 LED orange clignotantes 1500

Au-delà de Seuil 3 : 5 LED rouges clignotantes

Mesure Mise à jour

Sont réglables depuis l'application :

- \* **Durée (en s)** : durée totale des mesures
- \* **Intervalle (en s)** : durée entre deux mesures consécutives
- \* **Seuil 1** : Si la concentration en CO<sub>2</sub> < 800 ppm : 2 LED vertes allumées fixement
- \* **Seuil 2** : Si la concentration en CO<sub>2</sub> comprise entre 800 ppm et 1000 ppm : 2 LED orange allumées fixement
- \* **Seuil 3** : Si la concentration en CO<sub>2</sub> comprise entre 1000 ppm et 1500 ppm : 3 LED orange clignotent
- \* Et donc si la concentration en CO<sub>2</sub> > 1500 ppm : 5 LED rouges clignotent

- ✓ Appuyer sur « Mise à jour » (5 DEL vertes allumées) puis sur « Mesure »

- ✓ (Paramètres réglables depuis le programme Arduino

- \* le nombre de LED à allumer,
- \* le clignotement ou non des LED et la durée du clignotement,
- \* la couleur des LED.)

- ✓ À la fin d'une série de mesures, les 5 LED s'éteignent.
- ✓ Entre deux séries de mesures, débrancher/rebrancher l'Arduino et fermer/relancer l'application.
- ✓ En cas de soucis, redémarrer l'Arduino et relancer l'application.
- ✓ Des bugs de programmation persistent...

- **Affichage visuel grâce aux LED :**

- ✓ Si la concentration en CO<sub>2</sub> < 800 ppm : 2 LED vertes allumées fixement.
- ✓ Si la concentration en CO<sub>2</sub> comprise entre 800 ppm et 1000 ppm : 2 LED orange allumées fixement.
- ✓ Si la concentration en CO<sub>2</sub> comprise entre 1000 ppm et 1500 ppm : 3 LED orange clignotent.
- ✓ Si la concentration en CO<sub>2</sub> > 1500 ppm : 5 LED rouges clignotent.

- **Affichage des mesures sur le port série**

```

// capteur-co2-temperature-color - Arduino 1.8.13
// NDIR216.cpp - NDIR216.h

#include "NDIR216.h"
#include <SoftwareSerial.h>
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel pixels(5, 6, NEO_GRB + NEO_K8800);

SoftwareSerial mySerial(2, 3);
NDIR216 mySensor = NDIR216(mySerial);

SoftwareSerial bluetooth(10, 11); // (RX, TX) spin Re BT, pin Tx BT)

const unsigned long BLINK_PERIOD = 50;
const unsigned long BLINK_COUNTDOWN = 200;
//const unsigned long MEASURE_DELAY = 5000;

unsigned long duration = 6000000;
unsigned long interval = 50000;
long seuil1 = 900;
long seuil2 = 1000;
long seuil3 = 1300;

void Vert()
{
  pixels.clear();
  for(int i=0; i<2; i++) {
    pixels.setPixelColor(i, pixels.Color(0, 150, 0));
    pixels.show();
  }
  delay(interval);
}

void Rouge()
{
  pixels.clear();
  int nb_blink = interval / (2*BLINK_PERIOD);
  for (int j = 0; j < nb_blink; j++) {
    for(int i=0; i<5; i++) {
      pixels.setPixelColor(i, pixels.Color(150, 0, 0));
      pixels.show();
    }
    delay(BLINK_PERIOD);
    pixels.clear();
    pixels.show();
    delay(BLINK_PERIOD);
  }
}

void Bleu()
{
  for(int i=0; i<5; i++) {

```

- **Affichage des mesures sur la tablette ou smartphone**

Date	CO2	Temperature
15/06/2021 01:49:39	448 ppm	25 °C
15/06/2021 01:49:44	446 ppm	25 °C
15/06/2021 01:49:49	444 ppm	25 °C
15/06/2021 01:49:54	445 ppm	25 °C
15/06/2021 01:49:59	441 ppm	25 °C
15/06/2021 01:50:04	438 ppm	25 °C
15/06/2021 01:50:09	435 ppm	25 °C
15/06/2021 01:50:14	438 ppm	25 °C
15/06/2021 01:50:19	435 ppm	25 °C
15/06/2021 01:50:24	435 ppm	2

- **Fichiers csv**

- ✓ Icônes « gestionnaire de fichiers »
- ✓ Dossier « stockage »
- ✓ Les noms des fichiers de données : date\_heure de lancement des mesures

File Name	Size
15062021_02:28.csv	489 o
15062021_02:26.csv	485 o
15062021_02:25.csv	480 o
15062021_02:20.csv	357 o
15062021_02:17.csv	408 o