|  |  |
| --- | --- |
|  |  |

**PROJET QUALIT’AIR**

**DESCRIPTION ATELIER COLLEGE DE LA GUYONNERIE**

**MESURE DU TAUX DE CO2**

**1. Etapes préalables**

1.1. Informations CO2 – Qualité de l’air – COVID

1.2. Informations « Numérique ». Capteurs – Micro-controleurs – Programmation

**2. Maquettage sur platine de prototypage**

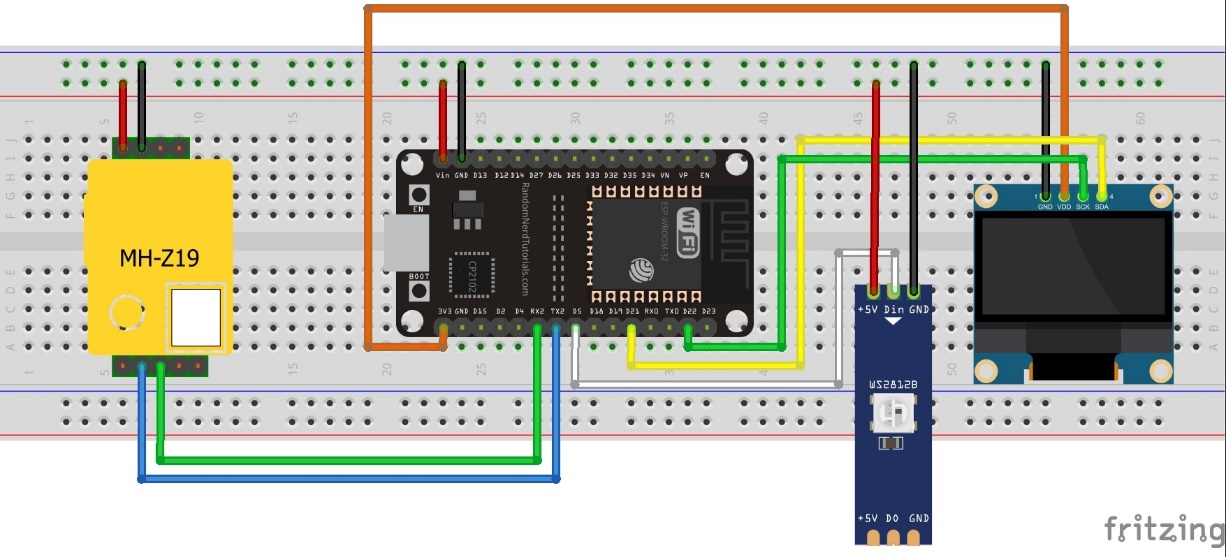
- Microcontrôleur : ESP32, doté de capacités Wifi, Bluetooth

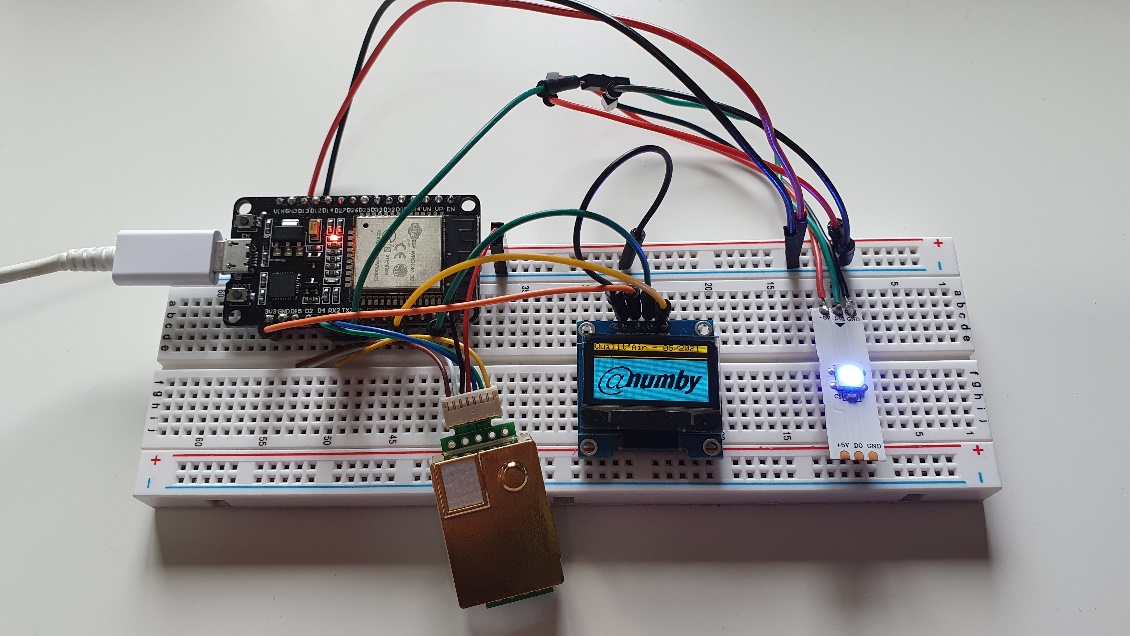
- Capteur CO2 de technologie NDIR (Non Dispersive Infra Red) réputée fiable et suffisamment précise pour nos usages (précision 50 ppm). Modèle MH-Z19 de chez Winsen.

- Afficheur écran OLED 0,96’’

- Led adressable (réglage couleur et luminosité par software)

- Alimentation USB 5V (type chargeur téléphone)

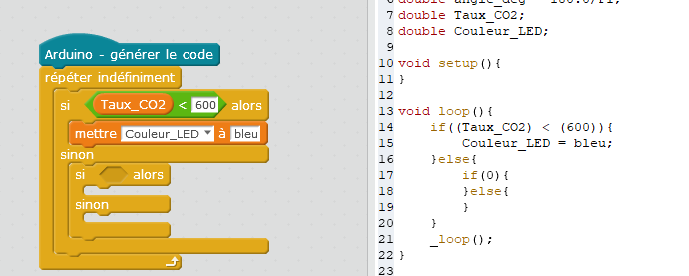




**3. Programmation - Code**

3.1. Codage mblock (base scratch orientée contrôle hardware)

Avec génération automatique du code Arduino / ESP, du type :

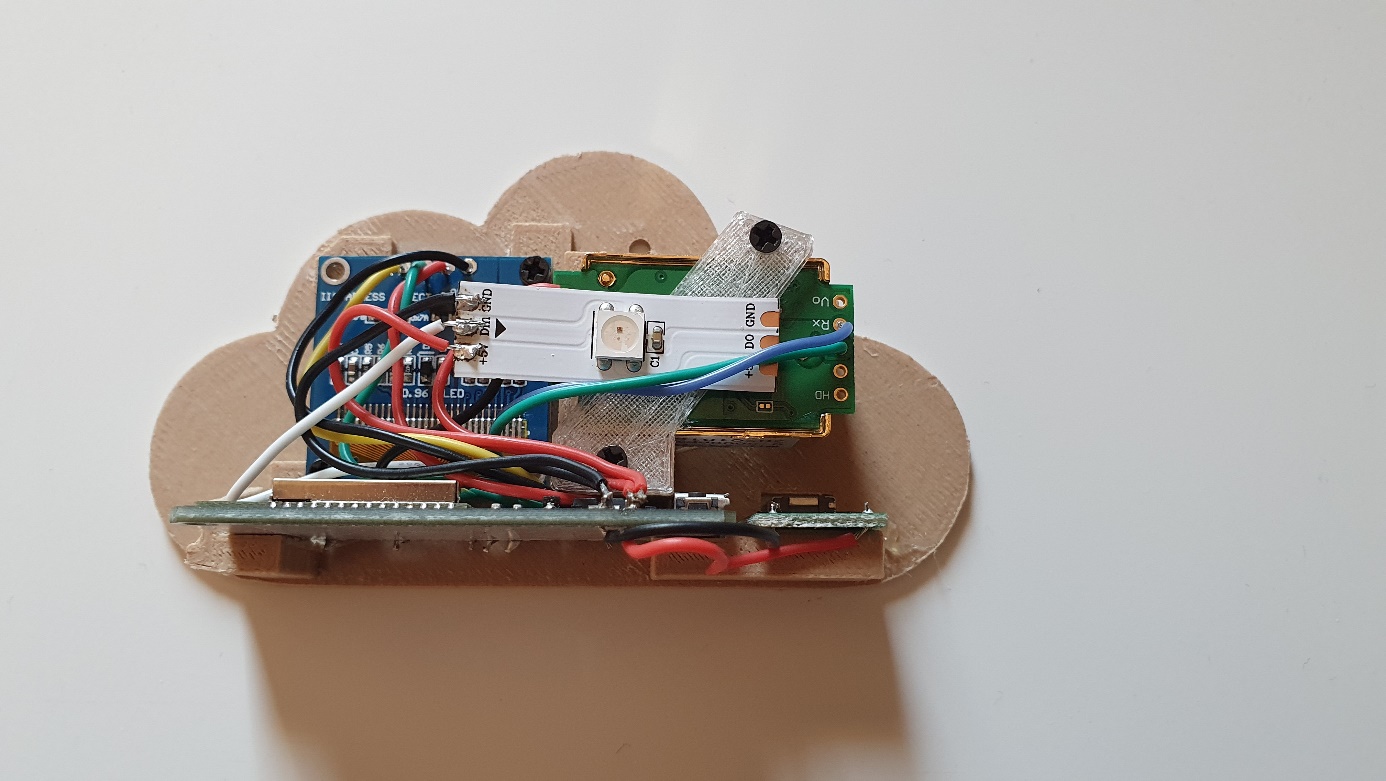


3.2. Codage C – IDE Arduino (cf. code en annexe)

3.3. Codage 100% mblock avec création d’extensions au préalable (gestion des bibliothèques, gestion des entrées sorties) ?

**4. Intégration – Impression 3D**







Version proposée par ANUMBY : Air’Cloud





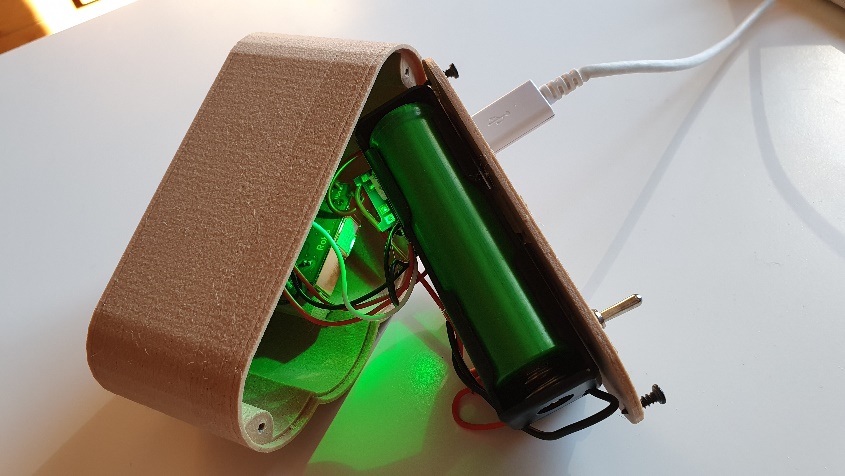
Variante :



|  |  |
| --- | --- |
|  |  |

**5. Evolutions – Améliorations fonctionnelles (après l’été 2021)**

5.1. Version autonome sur batterie :



Ajout d’une batterie Li-Ion 18650 avec contrôleur de charge.

Dispositif nomade autonome.

Fonctionnalité de calibration en « pleine nature »

5.2. Versions connectées

Exploitation des capacités WiFi ou Bluetooth du contrôleur ESP32 pour développer une version connectée.

Serveur Web intégré au device

Développement d’application smartphone

**ANNEXE : code**

/\*

\* Projet QUALIT'AIR

\* Modèle Air Cloud

\* Atelier collège

\* v01 (01.06.2021)

\* ======================================================

\* ESP32 30pin (DevKit V1)

\* Capteur MH-Z19B

\* Ecran OLED 0.96"

\* LED adressable WS2812B

\* ======================================================

\*

\*

\*/

#include <Arduino.h>

#include <Wire.h> // librairie bus I2C

#include <FastLED.h>

#define LED\_PIN 5

#define NUM\_LEDS 1

CRGB leds[NUM\_LEDS];

#include "MHZ19.h"

#define BAUDRATE 9600 // Device to MH-Z19 Serial baudrate (should not be changed)

MHZ19 myMHZ19; // Constructor for library

int taux\_CO2;

int8\_t Temp;

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// logo @numby (128x64px)

const unsigned char anumbyBitmap [] PROGMEM = {

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xfc, 0x0f, 0xc0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xf0, 0xff, 0xf8, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x81, 0xff, 0xff, 0xff,

0xff, 0xff, 0xc3, 0xff, 0xfe, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x83, 0xff, 0xff, 0xff,

0xff, 0xff, 0x0f, 0xff, 0xff, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x03, 0xff, 0xff, 0xff,

0xff, 0xfe, 0x3f, 0xff, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x07, 0xff, 0xff, 0xff,

0xff, 0xfc, 0x7f, 0xff, 0xff, 0x8f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x07, 0xff, 0xff, 0xff,

0xff, 0xf9, 0xff, 0xff, 0xff, 0x8f, 0x0f, 0xff, 0xff, 0xfe, 0x1e, 0x1e, 0x04, 0x7f, 0xff, 0xff,

0xff, 0xf3, 0xff, 0x87, 0xff, 0x80, 0x07, 0x03, 0x03, 0x00, 0x08, 0x0e, 0x00, 0x10, 0x38, 0x3f,

0xff, 0xe7, 0xfe, 0x01, 0x9f, 0x80, 0x03, 0x03, 0x03, 0x00, 0x00, 0x0e, 0x00, 0x18, 0x30, 0x7f,

0xff, 0xcf, 0xf8, 0x79, 0x3f, 0x00, 0x03, 0x07, 0x07, 0x00, 0x00, 0x0c, 0x00, 0x10, 0x30, 0xff,

0xff, 0x8f, 0xf1, 0xfc, 0x3f, 0x03, 0x06, 0x06, 0x06, 0x06, 0x04, 0x0c, 0x0c, 0x10, 0x60, 0xff,

0xff, 0x9f, 0xe7, 0xfc, 0x7f, 0x07, 0x06, 0x06, 0x06, 0x0e, 0x0c, 0x0c, 0x1c, 0x10, 0x61, 0xff,

0xff, 0x3f, 0xcf, 0xfc, 0x7e, 0x06, 0x06, 0x0e, 0x0e, 0x0c, 0x0c, 0x18, 0x1c, 0x10, 0x41, 0xff,

0xfe, 0x3f, 0x8f, 0xfc, 0xfe, 0x0e, 0x0c, 0x0c, 0x0c, 0x0c, 0x1c, 0x18, 0x38, 0x10, 0x43, 0xff,

0xfe, 0x7f, 0x1f, 0xf9, 0xfe, 0x0e, 0x0c, 0x1c, 0x0c, 0x18, 0x18, 0x18, 0x38, 0x10, 0x07, 0xff,

0xfe, 0x7f, 0x3f, 0xf9, 0xfc, 0x0c, 0x0c, 0x1c, 0x1c, 0x18, 0x18, 0x30, 0x38, 0x30, 0x07, 0xff,

0xfc, 0xfe, 0x3f, 0xf3, 0xfc, 0x1c, 0x18, 0x18, 0x18, 0x18, 0x30, 0x30, 0x70, 0x30, 0x0f, 0xff,

0xfc, 0xfe, 0x7f, 0xe3, 0xfc, 0x1c, 0x18, 0x38, 0x18, 0x30, 0x30, 0x30, 0x70, 0x70, 0x1f, 0xff,

0xfc, 0xfe, 0x7f, 0xe7, 0xf8, 0x18, 0x18, 0x38, 0x30, 0x30, 0x30, 0x60, 0x60, 0x70, 0x1f, 0xff,

0xf8, 0xfc, 0x7f, 0xcf, 0xf0, 0x38, 0x30, 0x30, 0x30, 0x30, 0x60, 0x60, 0x60, 0xf0, 0x3f, 0xff,

0xf9, 0xfc, 0x7f, 0x8f, 0xe0, 0x38, 0x30, 0x00, 0x30, 0x60, 0x60, 0x60, 0x01, 0xf0, 0x7f, 0xff,

0xf9, 0xfc, 0x7f, 0x0f, 0x80, 0x30, 0x30, 0x00, 0x60, 0x60, 0x60, 0xc0, 0x01, 0xf0, 0x7f, 0xff,

0xf9, 0xfc, 0x3c, 0x8e, 0x10, 0x70, 0x70, 0x00, 0x60, 0x60, 0xc0, 0xc0, 0x07, 0xf0, 0xff, 0xff,

0xf8, 0xfe, 0x01, 0x80, 0x7f, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xff, 0xfe, 0x1f, 0xe1, 0xff, 0xff,

0xf8, 0xff, 0x07, 0xc3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc1, 0xff, 0xff,

0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x03, 0xff, 0xff,

0xfc, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x07, 0xff, 0xff,

0xfe, 0x3f, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x0f, 0xff, 0xff,

0xfe, 0x1f, 0xff, 0xf8, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff,

0xff, 0x07, 0xff, 0xe1, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xc0, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xf0, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

};

// logo idf (128x64px)

const unsigned char idfBitmap [] PROGMEM = {

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xfd, 0xf3, 0xff, 0xc7, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xfe, 0xf3, 0xff, 0x92, 0x3f, 0xff, 0xff, 0xff, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xfe, 0x63, 0xff, 0xba, 0x3f, 0xff, 0xff, 0xff, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xfe, 0x03, 0xdf, 0xfe, 0x3f, 0xff, 0xff, 0xff, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0x02, 0x3f, 0xfe, 0x3e, 0x3f, 0xfb, 0xff, 0x1f, 0xff, 0x38, 0x7f, 0xc7, 0xf1, 0xfe, 0x3f,

0xe3, 0x00, 0x7f, 0xc6, 0x38, 0x0f, 0xfb, 0xff, 0x1f, 0xe2, 0x20, 0x18, 0x03, 0xc0, 0x78, 0x0f,

0xf0, 0x00, 0xff, 0xc6, 0x31, 0xc7, 0xfb, 0xff, 0x1f, 0xe0, 0x47, 0x98, 0x61, 0x8e, 0x31, 0xc7,

0xfc, 0x00, 0xff, 0xc6, 0x33, 0xe7, 0x03, 0x87, 0x00, 0x63, 0xff, 0x98, 0xf1, 0x9f, 0x33, 0xe7,

0xfe, 0x00, 0x7f, 0xc6, 0x31, 0xe2, 0xfb, 0x7b, 0x00, 0x63, 0xfe, 0x18, 0xf1, 0x9f, 0xe3, 0xc7,

0xfc, 0x00, 0x0f, 0xc6, 0x30, 0x02, 0xfb, 0x7b, 0x1f, 0xe3, 0xe1, 0x98, 0xf1, 0x9f, 0xe0, 0x07,

0xf8, 0x01, 0xff, 0xc6, 0x33, 0xfe, 0xfb, 0x07, 0x1f, 0xe3, 0xc7, 0x98, 0xf1, 0x9f, 0xe3, 0xff,

0xf0, 0x01, 0xff, 0xc6, 0x31, 0xfe, 0xfb, 0x7f, 0x1f, 0xe3, 0xc7, 0x98, 0xf1, 0x9e, 0x33, 0xff,

0xe7, 0x99, 0xff, 0xc6, 0x38, 0xc6, 0xfb, 0x7b, 0x1f, 0xe3, 0xc7, 0x18, 0xf1, 0x80, 0x71, 0xc7,

0xff, 0x9c, 0xff, 0xc6, 0x3c, 0x0f, 0x03, 0x87, 0x1f, 0xe3, 0xe0, 0x98, 0xf1, 0xc0, 0xf8, 0x0f,

0xff, 0x9e, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

};

void setup()

{

// Open serial communication

Serial.begin(9600);

Wire.begin();

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

delay(500);

display.clearDisplay();

// Draw bitmap on the screen

display.drawBitmap(0, 0, anumbyBitmap, 128, 64, 1);

display.setTextColor(0);

display.setTextSize(1);

display.setCursor(1, 0);

display.println("Qualit'Air - 05/2021");

display.setCursor(1, 8);

display.println("un projet de l'ANUMBY");

display.display();

display.setTextColor(WHITE);

Serial.println("Affichage logo Anumby");

delay(6000);

display.clearDisplay();

// Draw bitmap on the screen

display.drawBitmap(0, 0, idfBitmap, 128, 64, 1);

display.setTextColor(0);

display.setTextSize(1);

display.setCursor(5, 0);

display.println("avec le financement");

display.setCursor(30, 8);

display.println("de la region");

display.display();

display.setTextColor(WHITE);

Serial.println("Affichage logo IdF");

delay(6000);

display.clearDisplay();

display.drawBitmap(0, 0, idfBitmap, 128, 64, 1);

display.setTextColor(0);

display.setTextSize(1);

display.setCursor(5, 0);

display.println("prechauffage capteur");

display.setCursor(30, 8);

display.println("en cours ...");

display.display();

display.setTextColor(WHITE);

delay(2000);

FastLED.addLeds<WS2812, LED\_PIN, GRB>(leds, NUM\_LEDS);

//

Serial2.begin(BAUDRATE);

myMHZ19.begin(Serial2);

myMHZ19.autoCalibration(false); // Turn auto calibration ON (OFF AutoCalibration(false))

}

void loop()

{

taux\_CO2 = myMHZ19.getCO2(); // Request CO2 (as ppm)

Temp = myMHZ19.getTemperature(); // Request Temperature (as Celsius)

Serial.print("co2(MH-Z19): ");

Serial.print(taux\_CO2);

Serial.println();

//

if (taux\_CO2 > 300 && taux\_CO2 < 5000) {

display.clearDisplay();

display.setTextSize(1);

display.setCursor(0, 20);

display.print("Taux CO2 : ");

display.print(taux\_CO2);

display.print(" ppm");

display.setCursor(0, 40);

display.print("Temperature : ");

display.print(Temp);

display.print(" C");

display.display();

if (taux\_CO2 < 600){

leds[0] = CRGB(0, 0, 255);

}

else if (taux\_CO2 >= 600 &&taux\_CO2 < 800){

leds[0] = CRGB(0, 255, 0);

}

else if (taux\_CO2 >= 800 && taux\_CO2 < 1200){

leds[0] = CRGB(255, 153, 0);

}

else if (taux\_CO2 >= 1200){

leds[0] = CRGB(255, 0, 0);

}

FastLED.show();

}

delay(3000);

}