

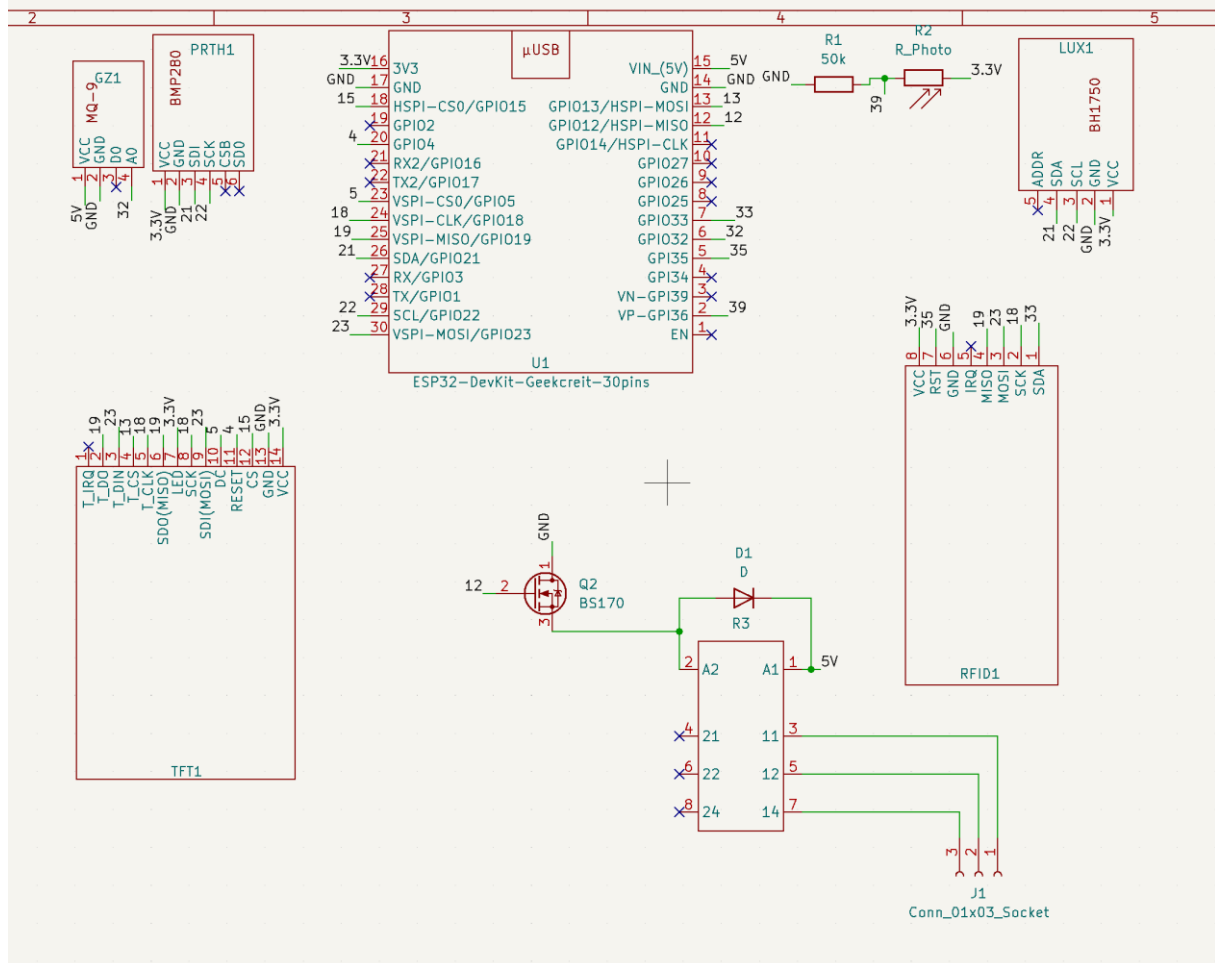
Dossier de fabrication
Lilian Carrière
BUT 2 GEII ESE A

Table des matières

I. Schéma structurel.....	3
II. Nomenclature des composants	4
III. Routage de la carte :	9
IV. Explication du programme de test du système	11
V. Procédure et protocole de test de la carte	16
VI. Fiche d'intervention	17
VII. Conclusion	23

I. Schéma structurel

Le schéma structurel de la carte électronique est le suivant :



Ce schéma permet de voir les connexions entre les différents composants et d'identifier les points sur lesquels intervenir en cas de besoins.

II. Nomenclature des composants

1) ESP32 :

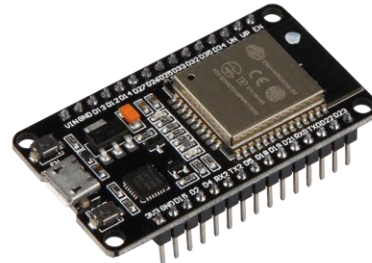
Référence : ESP32-DevKitC-32 ESP-WROOM-32

Dimensions du composant :

Largeur 28mm

Longueur : 51mm

Hauteur : 12mm



Prix : 3.09€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible d'utiliser un ESP8266 en remplacement de l'ESP32. Cependant, il sera nécessaire de revoir l'entièreté du câblage et de négliger certaines fonctionnalités.

2) Ecran tactile :

Référence : TFT TouchScreen ILI9341 2.8 pouces

Dimensions du composant :

Largeur : 42.72mm

Longueur : 77.18 mm

Hauteur : 15mm



Prix : 5.61€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible d'utiliser un écran similaire sans le tactile, ou alors d'utiliser un écran d'une dimension différente mais il faut alors changer la carte et le programme

3) BH1750 Light sensor :

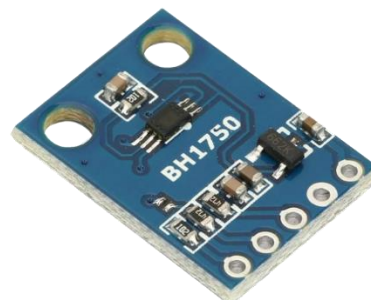
Référence : BH1750Light sensor

Dimensions du composant :

Largeur : 14.4mm

Longueur : 19.25mm

Hauteur : 10mm



Prix : 1.47€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible de remplacer le BH1750 par le VEML7700 disponible sur Ali Express pour 1.95€. Ce dernier présente des caractéristiques proches de celles du BH1750 et il fonctionne via communication I2C. Il suffit alors de rajouter la librairie associée à ce composant et à légèrement modifier le câblage de la carte.

4) BMP280:

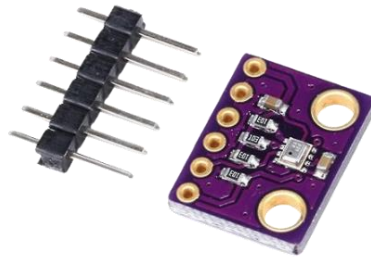
Référence : BMP280 Pressure and Temperature sensor

Dimensions du composant :

Largeur : 15mm

Longueur : 11mm

Hauteur : 10mm



Prix : 1.37€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il y a plusieurs composants qui peuvent aisément remplacer le BMP280. Il y a d'abord le BMP180, disponible sur Ali Express pour 0.25€, il s'agit de la version antérieure au BMP280. Ce composant offre les mêmes fonctionnalités que le 280 mais sa précision est moindre. Il y a aussi le BMP388, il s'agit d'une version plus récente du BMP280, il est plus précis que ce dernier mais son coût est bien plus élevé. Il est disponible sur Ali Express pour 3.77€. De plus, ce composant nécessitera de revoir le câblage de la carte pour le faire fonctionner.

5) RFID :

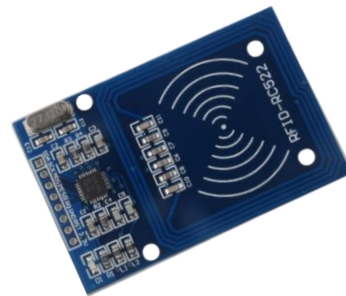
Référence : RFID-RC522

Dimensions du composant :

Largeur : 40mm

Longueur : 61mm

Hauteur : 14mm



Prix : 0.67€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible de remplacer le RFID-RC522 par le PN532 disponible sur Ali Express pour 0.10€. Ce composant offre l'avantage de pouvoir lire les puces RFID, mais aussi les puces NFC. Il fonctionne lui aussi via le protocole SPI.

6) Photorésistance :

Référence : Photorésistance LDR1000

Dimensions du composant :

Largeur : 4.7mm

Longueur : 5.66mm

Hauteur : 35mm

Prix : 1.70€

Site fournisseur : GO TRONIC

Quantité : 1

Composant de remplacement : Il est possible d'utiliser n'importe quelle autre photorésistance similaire, il faudra cependant changer le calcul présent dans le programme de la carte.



7) Résistance 50kΩ :

Référence : Résistance 50kΩ

Dimensions du composant :

Largeur : 3.91mm

Longueur : 2.1mm

Prix : 0.03€

Site fournisseur : Conrad

Quantité : 1

Composant de remplacement : Il est possible d'utiliser n'importe quelle résistance traversante de 50kΩ.



8) Relay finder 5V :

Référence : Finder 30.22.7.005.0000 5V/DC

Dimensions du composant :

Largeur : 10.6mm

Longueur : 20.8mm

Hauteur : 14mm

Prix : 4.29 €

Site fournisseur : Conrad

Quantité : 1

Composant de remplacement : Il est possible de remplacer le Finder 30.22.7.005.0000 5V/DC, par le Finder 40.52.005.0000 disponible chez Conrad pour 8.49€. Il dispose de caractéristiques similaires ou supérieures à celles du 30.22. Il nécessitera cependant une légère modification de la carte car il n'a pas les mêmes dimensions.



9) Transistor :

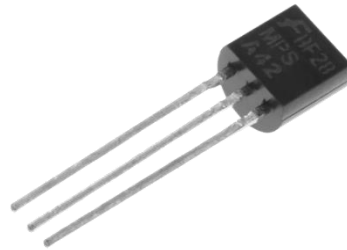
Référence : BS270

Dimensions du composant :

Largeur : 4.74mm

Longueur : 5.46mm

Hauteur : 10mm



Prix : 0.49€

Site fournisseur : Farnell

Quantité : 1

Composant de remplacement : Le BS270 est un transistor en canal N, de type CMOS, il est alors possible de le remplacer par n'importe quel autre transistor de même technologie qui présente des caractéristiques similaires.

10) MQ9 Gaz sensor :

Référence : MQ-9

Dimensions du composant :

Largeur : 20mm

Longueur : 32mm

Hauteur : 30mm



Prix : 1.06€

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible de remplacer le MQ-9 par le MQ-7 disponible chez Ali Express pour 1.07€. Il permet de détecter le monoxyde de carbone et le méthane comme le MQ-9. Il fonctionne aussi sur le même type de communication que le MQ-9.

11) Bornier à vis 1x3 :

Référence : KF301-3P

Dimensions du composant :

Largeur : 8mm

Longueur : 15.6mm

Hauteur : 14mm



Prix : 0.93€ (10 pièces)

Site fournisseur : Ali Express

Quantité : 1

Composant de remplacement : Il est possible de remplacer le KF301-3P par le WR-TBL 300VAC 10A 3P disponible sur Mouser pour 1.68€. Il permet de faire passer une tension allant jusqu'à 300V AC.

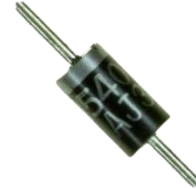
12) Diode :

Référence : 1N4004

Dimensions du composant :

Largeur : 3.2mm

Longueur : 12.86mm



Prix : 1.06€ (100 pièces)

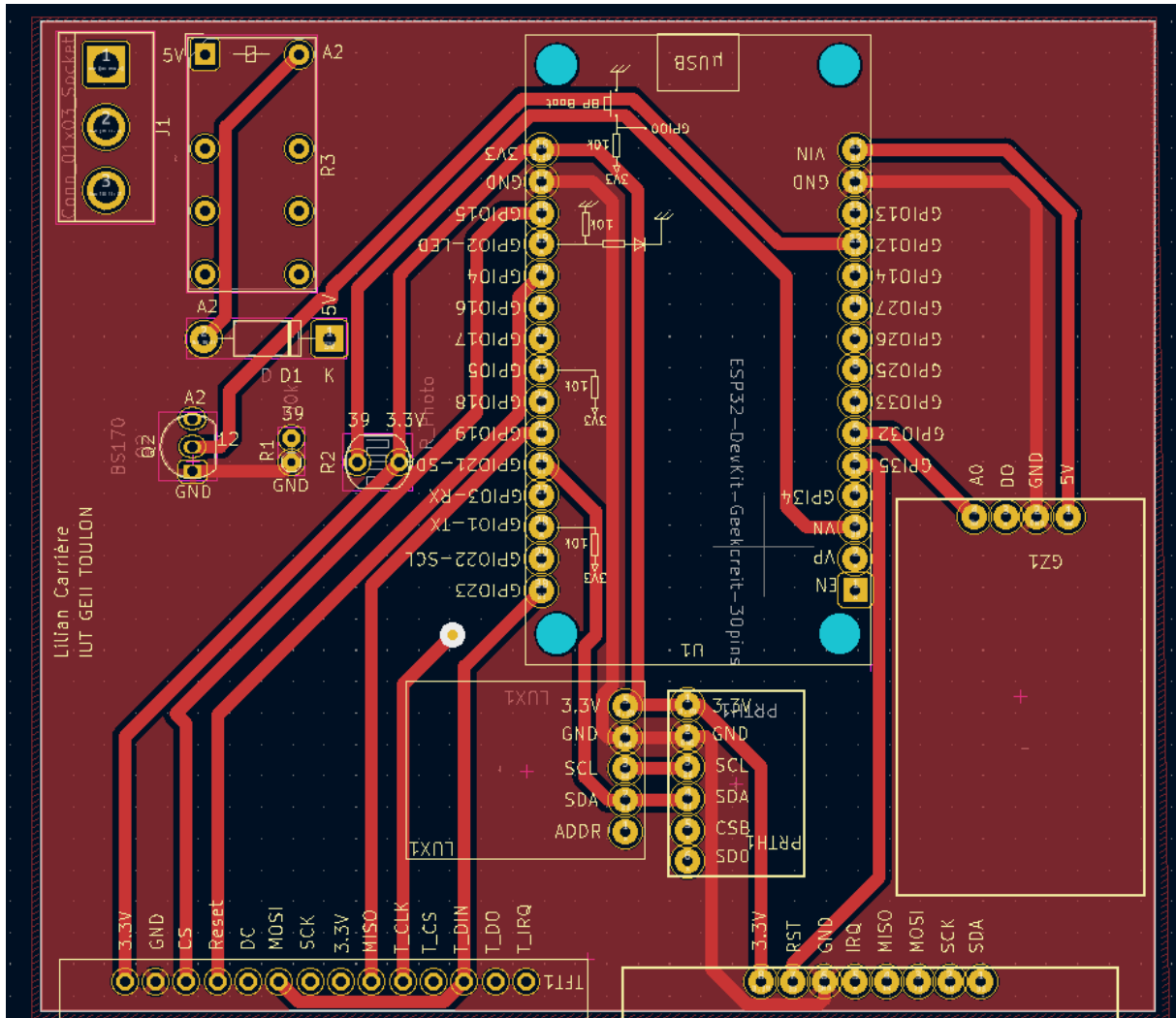
Site fournisseur : Ali Express

Quantité : 1

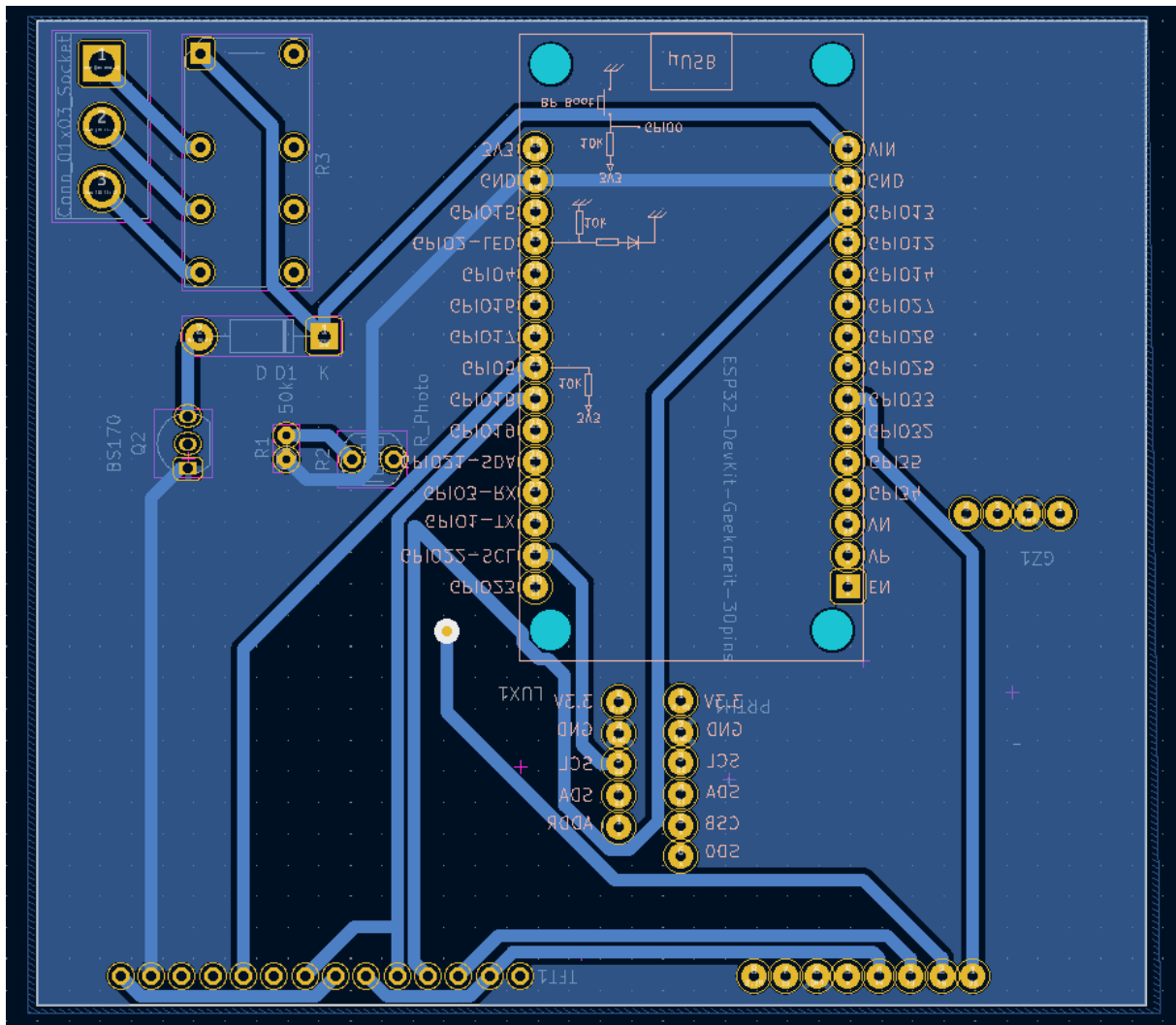
Composant de remplacement : Il est possible de remplacer cette diode par une autre diode de la série 1N400X car elles ont sensiblement les mêmes caractéristiques.

III. Routage de la carte :

Le routage de la carte a été effectué sur Kicad 7.0 et est le suivant :



Il s'agit de la face avant du PCB.



Il s'agit de la face arrière du PCB

Le routage de la carte sur Kicad permet de voir l'agencement réel des composants sur la carte. Tous les pins sont repérés par une écriture permettant au technicien de les localiser plus rapidement lors d'une intervention sur la carte.

La carte électronique possède les dimensions suivantes :

Hauteur : 80mm.

Largeur : 90mm.

Surface totale : 7200mm²

IV. Explication du programme de test du système

Le programme de test reste relativement simple afin de faciliter son utilisation par le technicien lors d'une intervention.

Cette partie du programme correspond à l'ajout des librairies qui sont nécessaires pour faire fonctionner les composants.

Il y a ensuite les différentes étapes de définition des composants et de leurs broches respectives. Tout d'abord, il faut associer un nom au BMP280 afin de pouvoir accéder aux fonctions de sa librairie. Ici ce sera « bmp ».

Il faut en faire de même pour le BH1750, ici son nom sera lighMeter.

Le transistor qui commande le relai est relié à la pin GPIO12 sur l'ESP32, on nomme donc la pin 12 Relai pour faciliter les opérations dans la suite du programme.

Il en va de même pour le MQ-9, sa broche A0 est reliée sur la broche GPIO32 de l'ESP32, on la renomme donc mqPin.

La méthode de définition de l'écran est presque similaire à celles du BMP et du BH. Il faut donner un nom à notre composant puis utiliser la fonction disponible dans la librairie pour lui associer ses fonctionnalités.

Il faut ensuite renseigner les paramètres du réseau utilisé, à savoir le nom du réseau ou SSID, et le mot de passe.

Enfin, il y a la phase d'initialisation des variables. Ces dernières seront utilisées dans le programme de test.

```
1 #include "WiFi.h"
2 #include "ESPAsyncWebServer.h"
3 #include <Wire.h>
4 #include <Adafruit_Sensor.h>
5 #include <Adafruit_BMP280.h>
6 #include <BH1750.h>
7 #include "SPIFFS.h"
8 #include <SPI.h>
9 #include <TFT_eSPI.h>
10
11
12 #define LED 2
13 /*-----BMP-----*/
14 Adafruit_BMP280 bmp;
15
16 /*-----BH-----*/
17 BH1750 lightMeter;
18
19 /*-----Relai-----*/
20 #define Relai 12
21
22 /*-----MQ-9-----*/
23 #define mqPin 32
24
25 /*-----Ecran-----*/
26 TFT_eSPI tft = TFT_eSPI();
27
28 /*-----WiFi-----*/
29 const char* ssid = "SFR_100F";
30 const char* password = "3hy17771umd9xx3q73pd";
31
32 /*-----Variables-----*/
33 char caractere = 0;
34 String msg, motcle;
35 int test = 0;
36 uint32_t now, last, last2;
37 bool LEDstate, RelaiState;
38 int etape, cpt;
39 int coValue;
40 float CoPPM;
```

```

void setup() {
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
  /*-----BMP-----*/
  bmp.begin(0x76);
  /*-----BH-----*/
  lightMeter.begin();
  /*-----Relai-----*/
  pinMode(Relai, OUTPUT);
  /*-----MQ-9-----*/
  pinMode(mqPin, INPUT);
  /*-----Ecran-----*/
  tft.init();
  SPI.begin();
  delay(2000);
  for (int j = 0; j < 10; j++) {
    Serial.println(" ");
  }
}

```

Après avoir renseigné les paramètres de configuration de chacun des composants, il y a le setup qui permet de mettre en marche les composants qui nécessitent une initialisation.

Il faut d'abord démarrer le port série et choisir la vitesse qui sera utilisée, ici elle est fixée à 115 200 bauds.

Puis, on configure la pin de la LED intégrée à l'ESP32 en tant que sortie afin de pouvoir l'utiliser comme telle.

L'initialisation du BMP s'effectue par l'utilisation d'une simple fonction « begin » dans laquelle on indique son adresse I2C.

Le BH1750 peut être initialisé de la même façon, il n'est pas obligatoire de préciser l'adresse I2C du composant.

Pour le relai, il faut définir la broche associée au transistor en tant que sortie. Et pour le MQ-9, il faut définir la broche associée à sa pin A0 en tant qu'entrée.

Le démarrage du TFT est fait grâce à la fonction « init » et au démarrage des fonctions de la librairie SPI.

```

delay(2000);
for (int j = 0; j < 10; j++) {
  Serial.println(" ");
}
Serial.println("-----Début du programme de test-----");
Serial.println("Pour accéder aux programmes de test de chacun des composants il faut taper les commandes entre '' suivantes : ");
Serial.println("MQ9 --> 'MQ9', affiche la concentration en PPM de monoxyde de carbone. (effectuer plusieurs fois les mesures car la valeur peut fortement varier entre deux mesures).");
Serial.println("BMP280 --> 'BMP', affiche l'adresse du capteur, la température et la pression.");
Serial.println("BH1750 --> 'BH', affiche l'adresse du capteur et le niveau de lumière.");
Serial.println("Relai --> 'RELAI', fait clavier le relai un certain nombre de fois.");
Serial.println("Ecran --> 'ECRAN', lance le calibrage de l'écran.");
Serial.println("Wifi --> 'WIFI', connecte la carte au Wifi, affiche le nom du réseau, le mot de passe et l'adresse IP donnée par le réseau.");
}

```

Cette partie du programme permet d'afficher les commandes à utiliser sur le moniteur série pour le technicien.

Le début du loop est composé des mises à jour de la variables now qui est utilisée pour les boucles de temps du programme. Mais aussi des variables LEDstate et RelaiState qui sont utilisées plus tard.

```

void loop() {
  now = millis();
  LEDstate = digitalRead(LED);
  RelaiState = digitalRead(Relai);
}

```

La boucle if suivante permet de faire clignoter la LED de l'ESP32 afin de montrer lorsque le programme est en route.

```

if (now - last >= 500) {
  digitalWrite(LED, !LEDstate);
  //Serial.println("Etat de la LED : " + String(LEDstate));
  last = now;
}

```

```

if (Serial.available() > 0) {

    caractere = Serial.read();

    //Serial.println("Caractère reçu : ");
    //Serial.println(caractere, DEC);

    if (caractere == '\n') {
        //Serial.println("Msg : " + msg);
        motcle = msg;
        msg = "";

        if (motcle == "MQ9") {
            Serial.println("Interface de test du MQ-9");
            test = 1;
        }

        if (motcle == "BMP") {
            Serial.println("Interface de test du BMP280");
            test = 2;
        }

        if (motcle == "BH") {
            Serial.println("Interface de test du BH1750");
            test = 3;
        }
        if (motcle == "RELAI") {
            Serial.println("Interface de test du Relai");
            Serial.println("-----");
            Serial.println("Programme de test du Relai en cours...");
            Serial.println("Le relai devrait claquer 10 fois à un interval plus ou moins régulier");
            test = 4;
        }
        if (motcle == "ECRAN") {
            Serial.println("Interface de test de l'écran");
            test = 5;
        }
        if (motcle == "WIFI") {
            Serial.println("Interface de test de la connectivité Wifi");
            test = 6;
        }
    } else {
        msg += caractere;
    }
}

```

Cette longue boucle if permet quant à elle de lire les instructions envoyées dans le moniteur série.

Son fonctionnement est simple : elle récupère chaque caractère qui est inscrit dans le port série et les entassent dans une variable msg jusqu'à la réception du caractère '\n'.

Une fois ce caractère reçu, le contenu de la variable msg est transféré dans la variable motcle. msg qui est ensuite vidée.

Le contenu de motcle est comparé afin de pouvoir rediriger le programme vers le test du composant désiré.

Si le technicien entre 'MQ9', le programme entrera alors dans cette boucle permettant d'afficher la concentration en PPM du monoxyde de carbone aux alentours du MQ9. Puis il revient dans la boucle principale.

```

if (test == 1) {
    Serial.println("-----");
    Serial.println("Programme de test du MQ9 en cours...");
    coValue = analogRead(mqPin);
    float a = 599.68, b = -2.244;
    CoPPM = a * pow(coValue, b);
    Serial.println("Concentration en PPM de monoxyde de carbone : " + String(CoPPM));

    Serial.println("Programme de test du MQ9 terminé");
    Serial.println("-----");
    test = 0;
}

```

```

if (test == 2) {
    Serial.println("-----");

    Serial.println("Programme de test du BMP280 en cours...");

    ScanBMP();
    Serial.print("Température = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" °C");

    Serial.print("Pression = ");
    Serial.print(bmp.readPressure() / 100.0F); // Conversion de la pression en hPa
    Serial.println(" hPa");
    Serial.println("Programme de test du BMP280 terminé");
    Serial.println("-----");

    test = 0;
}

```

Si le technicien entre 'BMP', le programme entre alors dans la boucle suivante. Elle effectue un scan des adresses I2C présentes sur les broches de l'ESP32 et renvoie celle du BMP280. Ensuite, elle affiche la température et la pression au niveau du capteur. Enfin, elle revient dans la boucle principale.

Si le technicien entre 'BH', le programme entre dans la boucle suivante. Elle effectue un scan des adresses I2C présentes sur les broches de l'ESP32 et renvoie celle du BH1750. Puis elle affiche le niveau d'intensité lumineuse autour du capteur et retourne dans la boucle principale.

```
if (test == 3) {
    Serial.println("-----");

    Serial.println("Programme de test du BH1750 en cours...");

    ScanBH();
    Serial.println("Intensité lumineuse : " + String(lightMeter.readLightLevel()) + "lux");

    Serial.println("Programme de test du BH1750 terminé");
    Serial.println("-----");

    test = 0;
}
```

```
if (test == 4) {

    if (now - last2 >= 500) {
        digitalWrite(Relai, !RelaiState);
        last2 = now;
        cpt++;
    }

    if (cpt >= 10) {
        test = 0;
        cpt = 0;
        Serial.println("Programme de test du Relai terminé");
        Serial.println("-----");
    }
}
```

Si le technicien entre 'RELA', le programme entre dans la boucle suivante. Celle-ci contient une boucle de temps permettant de faire commuter le relai à une fréquence de deux changements d'états par seconde. Il effectue cette boucle dix fois. Il change donc dix fois d'état. Puis le programme retourne dans la boucle principale.

```
if (test == 5) {
    Serial.println("-----");

    Serial.println("Programme de test du Ecran en cours...");
    calibrage();
    Serial.println("Programme de test du Ecran terminé");
    Serial.println("-----");

    test = 0;
}
```

Si le technicien entre 'ECRAN', le programme entre dans la boucle suivante. Cette boucle lance le programme de calibrage de l'écran TFT. Le technicien doit alors compléter cette séquence de calibrage. Une fois terminée, le

programme retourne dans la boucle principale.

Enfin, si le technicien entre 'WIFI', le programme entre dans la boucle suivante. Cette boucle initialise le module Wifi de l'ESP32 et se connecte au réseau sélectionné avant le setup. Il affiche si l'opération a réussi et par la suite le nom du réseau, le mot de passe et l'adresse IP que l'ESP32 a obtenu. Le programme retourne ensuite dans la boucle principale.

```
if (test == 6) {
    Serial.println("-----");

    Serial.println("Programme de test du WiFi en cours...");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi");
    Serial.println("Nom du réseau : " + String(ssid));
    Serial.println("Mot de passe du réseau : " + String(password));
    Serial.println(WiFi.localIP());
    Serial.println("Programme de test du WiFi terminé");
    Serial.println("-----");

    test = 0;
}
```

Puis, il y a les fonctions qui sont utilisées lors des tests.

```

void ScanBMP() {
  for (byte i = 8; i < 120; ++i) {
    Wire.beginTransaction(i);
    byte error = Wire.endTransmission();
    if (error == 0) {
      if (i == 0x76) {
        Serial.print("Adresse du BMP : 0x");
        if (i < 16) Serial.print("0");
        Serial.println(i, HEX);
      }
    }
  }
}

```

Il y a d'abord la fonction qui permet de renvoyer l'adresse du BMP280. Elle démarre un balayage d'adresse possible puis si elle obtient une réponse de la part de l'adresse 0x76, elle indique que c'est celle appartenant au BMP280 sur le port série.

```

void ScanBH() {
  for (byte i = 8; i < 120; ++i) {
    Wire.beginTransaction(i);
    byte error = Wire.endTransmission();
    if (error == 0) {
      if (i == 0x23) {
        Serial.print("Adresse du BH : 0x");
        if (i < 16) Serial.print("0");
        Serial.println(i, HEX);
      }
    }
  }
}

```

Le fonctionnement de la fonction qui retourne l'adresse du BH1750 est la même que celle du BMP280, sauf qu'au lieu d'attendre une réponse de l'adresse 0x76, elle en attend une de la part de l'adresse 0x23.

La dernière fonction est le calibrage de l'écran TFT. Elle commence par afficher un fond noir, puis elle affiche le texte « Touchez l'écran a » en partant du point se trouvant aux coordonnées 25 : 75.

Elle affiche ensuite le texte « chaque coin indique. » en revenant à la ligne.

Puis elle affiche une flèche bleue dans un des coins en attendant que le coin soit touché. Une fois touché, la flèche bleue se déplace dans un autre coin et attend de nouveau d'être touchée. L'opération se répète jusqu'à ce que tous les coins aient été touchés. Une fois que c'est le cas, le texte « Calibration terminée ! » est affiché en vert en dessous des autres textes.

```

void calibrage() {
  uint16_t calData[5];
  uint8_t calDataOK = 0;

  tft.fillScreen(TFT_BLACK);
  tft.setCursor(25, 70);
  tft.setTextFont(2);
  tft.setTextSize(2);
  tft.setTextColor(TFT_WHITE, TFT_BLACK);
  tft.println("Touchez l'ecran a ");
  tft.setCursor(15, 110);
  tft.println("chaque coin indique.");
  tft.setTextFont(1);
  tft.println();
  tft.calibrateTouch(calData, TFT_YELLOW, TFT_BLACK, 20);
  tft.setTextColor(TFT_GREEN, TFT_BLACK);
  tft.println("Calibration terminee!");
}

```

V. Procédure et protocole de test de la carte

Pour que le technicien puisse intervenir sur la carte et qu'il soit le plus efficace possible, il faut qu'il suive un plan d'intervention.

Lorsque le technicien doit effectuer des tests sur une carte électronique de ce type, il doit d'abord se pourvoir de sa fiche d'intervention qui est donnée plus bas dans le dossier. Puis il doit se munir d'un ordinateur contenant le programme de test de la carte, ainsi que des appareils suivants :

- 1 Multimètre
- 1 Thermomètre
- 1 Luxmètre
- 1 Baromètre
- 1 Détecteur de monoxyde de carbone
- 1 Source d'alimentation stabilisée

Il doit également se munir d'une source de lumière dont l'intensité est modulable, d'au minimum une puce RFID, d'une source de chaleur modulable, et de quoi générer de petite quantité de monoxyde de carbone.

Une fois que le technicien est équipé, il peut commencer par vérifier l'intégrité physique de l'ensemble de la carte et de chacun des composants. S'il repère un ou plusieurs défauts, il doit l'indiquer dans la rubrique 1 de la fiche d'intervention.

Après avoir inspecté la carte il peut alors la brancher à son ordinateur ou à une source d'alimentation stabilisée. Il peut ainsi vérifier si la LED présente sur l'ESP32 s'allume. Si ce n'est pas le cas, il doit l'inscrire dans la rubrique 1 de la fiche d'intervention et procéder au remplacement de l'ESP32.

Une fois la carte branchée et mise sous tension, il peut donc vérifier les tensions d'alimentation de chacun des composants en comparant avec les valeurs de la rubrique 2 de la fiche d'intervention. Il peut aussi procéder aux tests de continuités conformément à la rubrique 3 de cette même fiche. Après avoir terminé, il peut alors téléverser le programme de test Arduino fourni. Une fois le programme installé dans la carte, il procède aux essais de chacun des composants via le moniteur série intégré d'Arduino.

La procédure des tests relatifs à chacun des composants est indiquée sur la rubrique 4 de la fiche d'intervention.

VI. Fiche d'intervention

Test	Désignation	Résultat			Date	Nom
		Min	Valeur	Max		
1	Intégrité physique - Carte électronique - ESP32 - BMP 280 - BH1750 - Ecran TFT - MQ-9 - Relai - RFID - Bornier à vis - Photorésistance - Transistor - Résistance - Diode 1N4004		OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK			
2	Tension d'alimentation - ESP32 - BMP280 - BH1750 - Ecran TFT - MQ-9 - Relai - RFID - Photorésistance - Transistor - Résistance - Diode 1N4004	2.7 V 1.71 V 2.4 V 2.7 V 4.8 V 4.75 V 3.3 V 3.1 V 4.8 V 3.1 V 4.8 V		3.6 V 3.6 V 3.6 V 5V 5.2 V 5.25 V 3.6 V 3.5 V 5.2 V 3.1 V 5.2V		

3	Tests de continuité					
3.1	BMP280 : - Conductivité entre pin 3.3V BMP et pin 3.3V ESP32 - Conductivité entre pin GND BMP et pin GND ESP32 - Conductivité entre pin SCL BMP et GPIO22 ESP32 - Conductivité entre pin SDA BMP et GPIO21 ESP32		OK/NOK OK/NOK OK/NOK OK/NOK			
3.2	BH1750 : - Conductivité entre pin 3.3V BH et pin 3.3V ESP32 - Conductivité entre pin GND BH et pin GND ESP32 - Conductivité entre pin SCL BH et pin GPIO22 ESP32 - Conductivité entre pin SDA BH et pin GPIO21 ESP32		OK/NOK OK/NOK OK/NOK OK/NOK			
3.3	MQ-9 : - Conductivité entre pin 5V MQ et pin VIN ESP32 - Conductivité entre pin GND MQ et pin GND ESP32 - Conductivité entre pin A0 MQ et pin GPIO32 ESP32		OK/NOK OK/NOK OK/NOK			
3.4	RFID : - Conductivité entre pin 3.3V RFID et pin 3.3V ESP32 - Conductivité entre pin RST RFID et pin GPIO35 ESP32 - Conductivité entre pin GND RFID et pin GND ESP32 - Conductivité entre pin MISO RFID et pin GPIO19 ESP32 - Conductivité entre pin MOSI RFID et pin GPIO23 ESP32		OK/NOK OK/NOK OK/NOK OK/NOK OK/NOK			

3.5	- Conductivité entre pin SCK RFID et pin GPIO18 ESP32	OK/NOK			
	- Conductivité entre pin SDA RFID et pin GPIO33 ESP32	OK/NOK			
3.5	Ecran TFT :				
	- Conductivité entre pin 3.3V TFT et pin 3.3V ESP32	OK/NOK			
	- Conductivité entre pin GND TFT et pin GND ESP32	OK/NOK			
	- Conductivité entre pin CS TFT et pin GPIO15 ESP32	OK/NOK			
	- Conductivité entre pin RESET TFT et pin GPIO4 ESP32	OK/NOK			
	- Conductivité entre pin DC TFT et pin GPIO5 ESP32	OK/NOK			
	- Conductivité entre pin MOSI TFT et pin GPIO23 ESP32	OK/NOK			
	- Conductivité entre pin SCK TFT et pin GPIO18 ESP32	OK/NOK			
	- Conductivité entre pin LED TFT et pin 3.3V ESP32	OK/NOK			
	- Conductivité entre pin MISO TFT et pin GPIO19 ESP32	OK/NOK			
	- Conductivité entre pin T_CLK TFT et pin GPIO18 ESP32	OK/NOK			
	- Conductivité entre pin T_CS TFT et pin GPIO13 ESP32	OK/NOK			
	- Conductivité entre pin T_DIN TFT et pin GPIO23 ESP32	OK/NOK			
	- Conductivité entre pin T_DO TFT et pin GPIO19 ESP32	OK/NOK			
3.6	Relai :				
	- Conductivité entre pin 5V Relai et pin VIN ESP32	OK/NOK			
	- Conductivité entre pin A2 Relai et pin A2 Diode 1N4004	OK/NOK			
	- Conductivité entre pin 11 Relai et pin 11 Bornier	OK/NOK			
	- Conductivité entre pin 12 Relai et pin 12 Bornier	OK/NOK			

3.7	- Conductivité entre pin 14 Relai et pin 14 Bornier		OK/NOK			
	Transistor :					
	- Conductivité entre pin GND Transistor et pin GND ESP32		OK/NOK			
	- Conductivité entre pin A2 Transistor et pin A2 Diode 1N4004		OK/NOK			
	- Conductivité entre pin 12 Transistor et GPIO 12 ESP32		OK/NOK			
	3.8	Photorésistance :				
	- Conductivité entre pin 39 Photorésistance et pin VN ESP32		OK/NOK			
	- Conductivité entre pin 3.3V Photorésistance et pin 3.3V ESP32		OK/NOK			
	3.9	Résistance :				
	- Conductivité entre pin 39 Résistance et pin 39 Photorésistance		OK/NOK			
3.10	- Conductivité entre pin GND Résistance et pin GND ESP32		OK/NOK			
	Diode 1N4004 ;					
	- Conductivité entre pin 5V Diode 1N4004 et pin VIN ESP32		OK/NOK			
3.11	- Conductivité entre pin A2 Diode 1N4004 et pin A2 Relai		OK/NOK			
	Bornier à vis :					
	- Conductivité entre pin 11 Bornier et pin 11 Relai		OK/NOK			
	- Conductivité entre pin 12 Bornier et pin 12 Relai		OK/NOK			
	- Conductivité entre pin 14 Bornier et pin 14 Relai		OK/NOK			

4	Test des composants					
4.1	BMP280 : <ul style="list-style-type: none"> - Valeurs de température et de pression affichées sur le moniteur série - Valeur de température semblable à celle du thermomètre - Valeur de pression semblable à celle du baromètre - Réactivité au changement de température 		OK/NOK OK/NOK OK/NOK OK/NOK			
4.2	BH1750 : <ul style="list-style-type: none"> - Intensité lumineuse affichée sur le moniteur série - Intensité lumineuse semblable a celle indiquée par le luxmètre - Réactivité au changement de luminosité 		OK/NOK OK/NOK OK/NOK			
4.3	MQ-9 : <ul style="list-style-type: none"> - Concentration en monoxyde affiché sur le moniteur série - Concentration semblable à celle du détecteur de monoxyde - Réactivité au changement de concentration du monoxyde 		OK/NOK OK/NOK OK/NOK			
4.4	RFID : <ul style="list-style-type: none"> - Connexion avec l'ESP32 réussie - Détection de la puce RFID - Affichage de l'ID de la puce 		OK/NOK OK/NOK OK/NOK			
4.5	Ecran TFT :					

	<ul style="list-style-type: none"> - Connexion avec l'ESP32 réussie - Affichage séquence calibrage sur l'écran - Possibilité de faire le calibrage - Affichage réussite sur l'écran 		OK/NOK			
			OK/NOK			
			OK/NOK			
			OK/NOK			
4.6	Relai :					
	<ul style="list-style-type: none"> - Commutation du relai périodiquement 		OK/NOK			
4.7	Transistor :					
	<ul style="list-style-type: none"> - Commutation du relai périodiquement 		OK/NOK			
4.8	Photorésistance :					
	<ul style="list-style-type: none"> - Intensité lumineuse affichée sur le moniteur série 		OK/NOK			
	<ul style="list-style-type: none"> - Intensité lumineuse semblable à celle du luxmètre 		OK/NOK			
	<ul style="list-style-type: none"> - Réactivité au changement de luminosité 		OK/NOK			

VII. Conclusion

En conclusion, ce dossier permet de transmettre toutes les informations nécessaires à un technicien qui ne connaît rien de la carte mais devant effectuer une intervention dessus. Cela lui permettra d'être en mesure de réaliser des essais et de remplacer les composants présents sur celle-ci si cela se trouve être nécessaire.