

1 Bibliothèques

cpp

 Copier le code

```
#include <WiFi.h>           // Pour se connecter au WiFi
#include <WiFiManager.h>    // Pour configurer facilement le WiFi chez toi
#include <Adafruit_GFX.h>    // Pour dessiner sur l'écran OLED
#include <Adafruit_SSD1306.h> // Pour contrôler l'écran OLED SSD1306
#include <Adafruit_NeoPixel.h> // Pour gérer les LEDs NeoPixel
#include "time.h"           // Pour récupérer l'heure via Internet (NTP)
```

Explications :

- Ces bibliothèques ajoutent les fonctionnalités de WiFi, gestion OLED et LEDs.
- À installer depuis le **Gestionnaire de bibliothèques Arduino** si ce n'est pas déjà fait.

2 Paramètres WiFi (WiFiManager)

cpp

 Copier le code

```
WiFiManager wifiManager; // Création d'un objet WiFiManager
```

Explications :

- WiFiManager permet à l'élève de connecter la boîte à son WiFi maison sans toucher au code.
- Si le WiFi n'est pas configuré → la boîte crée un réseau "Reveil-xxxx" pour choisir le WiFi.

3 Paramètres OLED

cpp

 Copier le code

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

Explications :

- Définition de la taille de l'écran.
- `display` sert à écrire du texte et afficher l'heure.

4 Paramètres LED NeoPixel

cpp

 Copier le code

```
#define LED_PIN 5
#define NUMPIXELS 7
Adafruit_NeoPixel pixels(NUMPIXELS, LED_PIN, NEO_GRB + NEO_KHZ800);
```

Explications :

- `LED_PIN` = broche connectée aux LEDs
- `NUMPIXELS` = nombre de LEDs
- `NEO_GRB` = ordre des couleurs attendu par les NeoPixel

5 Couleur personnalisée

cpp

 Copier le code

```
uint8_t colorR = 0; // Rouge (0-255)
uint8_t colorG = 255; // Vert (0-255)
uint8_t colorB = 0; // Bleu (0-255)
```

Explications :

- L'élève peut modifier ces valeurs pour changer la couleur de la LED.
- Après changement, il faut re-téléverser le code pour appliquer la nouvelle couleur.

6 Configuration dans `setup()`

cpp

 Copier le code

```
void setup() {
  Serial.begin(115200);

  // Démarrage WiFiManager pour configurer Le WiFi
  wifiManager.autoConnect("Reveil-xxxx"); // Si pas de WiFi connu, crée un réseau temporaire

  // Initialisation écran OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("Échec OLED");
    for(;;); // Stop si OLED non détecté
  }
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("Init OK");
  display.display();
  delay(2000);

  // Initialisation LEDs
  pixels.begin();
  pixels.clear();
  pixels.show();

  // Configuration NTP
  const long gmtOffset_sec = 3600;
  const int daylightOffset_sec = 3600;
  configTime(gmtOffset_sec, daylightOffset_sec, "pool.ntp.org");
}
```

Explications :

- `autoconnect` crée un réseau temporaire si pas de WiFi connu.
- L'écran OLED affiche "Init OK" pour confirmer que tout fonctionne.
- LEDs et heure Internet sont prêtes à fonctionner.

7 Fonction pour afficher l'heure

cpp

 Copier le code

```
void showTime() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) return; // Rien si pas de WiFi

    char buffer[16];
    strftime(buffer, sizeof(buffer), "%H:%M:%S", &timeinfo);

    display.clearDisplay();
    display.setCursor(10, 25);
    display.print(buffer);
    display.display();
}
```

Explications :

- Récupère l'heure via NTP et l'affiche sur l'OLED.
- Si pas de WiFi → la fonction ne fait rien.

8 Boucle principale

cpp

 Copier le code

```
void loop() {
    showTime(); // Met à jour L'heure

    // LEDs couleur personnalisée
    for (int i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(colorG, colorR, colorB)); // ordre GRB
    }
    pixels.show();

    delay(1000); // Met à jour chaque seconde
}
```

Explications :

- L'heure est mise à jour chaque seconde.
- Les LEDs affichent la couleur choisie par l'élève.
- L'ordre GRB est important pour que la couleur s'affiche correctement.