

# Cahier des Charges – Projet IoT

*(Arduino Mega → LoRa SX1278 → ESP32 → MQTT → Dashboard)*

## AUTHORS

- Louis GAMBART
- Guillaume MECENE

## 1. Contexte du projet

Le projet vise à concevoir un système complet de surveillance environnementale permettant :

- la **mesure locale** de différents paramètres (humidité du sol, température, luminosité...),
- l'**affichage sur un écran LCD**,
- la **transmission longue portée via LoRa**,
- l'envoi des données vers un **serveur MQTT** via ESP32,
- l'exploitation des données dans un **dashboard externe**.

## 2. Objectifs généraux

1. Monitorer l'environnement d'une plante ou d'un espace intérieur.
2. Afficher les données localement sur un écran LCD.
3. Fournir un canal de communication LoRa longue portée.
4. Acheminer les données via MQTT vers un système de visualisation.
5. Fournir une architecture robuste, modulaire, évolutive et sécurisée.

## 3. Périmètre du projet

### Phase 1 – Système local (Arduino Mega)

- Lecture de tous les capteurs.
- Logique d'affichage LCD avec extinction automatique.
- Gestion d'un bouton de commande.
- Signalisation via LEDs.
- Code fonctionnel complet sur Arduino.

## **Phase 2 – Système connecté**

- Ajout d'un module LoRa SX1278 sur l'Arduino.
- Réception LoRa sur un ESP32.
- Transmission vers un broker MQTT.
- Interface utilisateur via un dashboard.

## **4. Description matérielle**

### **4.1 Composants Arduino Mega (nœud capteurs)**

- **Arduino Mega 2560**
- **FC-28** : humidité du sol + module analogique
- **DHT22** : température + humidité de l'air
- **TSL2591** : capteur de luminosité (bus I<sup>2</sup>C)
- **LEDs** : rouge (alerte), verte (normal), bleue (LCD actif)
- **Bouton poussoir** (lecture manuelle / activation LCD)
- **Écran LCD 16×2 ou 20×4** en mode parallèle
- Breadboards + câblage

### **4.2 Passerelle LoRa/MQTT (ESP32)**

- **ESP32 DevKit v1**
- **Module LoRa SX1278**
- Connexion WiFi pour MQTT

## **5. Exigences fonctionnelles – Phase 1 (local)**

### **F1 – Lecture automatique des capteurs**

Toutes les **60 secondes** :

- humidité du sol (FC-28),
- luminosité (TSL2591),
- température (DHT22),
- humidité relative de l'air (DHT22).

### **F2 – Lecture manuelle via le bouton**

Quand le bouton est appuyé :

1. Lecture immédiate des capteurs,
2. Mise à jour instantanée de l'affichage,
3. Activation du LCD pour **10 secondes**,

4. LED bleue activée en même temps que le LCD.

Anti-rebond obligatoire.

### **F3 – Gestion du rétroéclairage du LCD**

- LCD éteint par défaut,
- Allumage lorsque le bouton est pressé,
- Timer de **10 secondes**, réinitialisé à chaque appui,
- Extinction automatique (LCD + LED bleue).

### **F4 – Gestion des LEDs**

- **LED bleue** : ON lorsque le LCD est actif.
- **LED verte** : conditions normales.
- **LED rouge** : alerte (sol sec, erreur capteur, valeurs anormales).

### **F5 – Affichage LCD**

Lorsque l'écran est allumé, afficher :

- Température,
- Humidité de l'air,
- Humidité du sol,
- Luminosité,
- Message d'état (« Sol sec », « Luminosité faible », etc.).

## **6. Exigences fonctionnelles – Phase 2 (LoRa / MQTT)**

### **F6 – Envoi LoRa (Arduino)**

Toutes les 60 secondes ou lors d'une lecture manuelle, envoyer un paquet LoRa contenant :

- `soil` : humidité du sol,
- `temp` : température,
- `hum` : humidité de l'air,
- `lux` : luminosité,
- `state` : état (OK, ALERTE, ERREUR),
- `t` : timestamp ou numéro de trame.

Format JSON recommandé :

```
{"soil":23,"temp":21.4,"hum":55,"lux":1200,"state":0,"t":1025}
```

## **F7 – Réception LoRa sur ESP32**

L'ESP32 doit :

- recevoir les messages LoRa,
- vérifier les données,
- parser les valeurs,
- préparer un envoi MQTT.

## **F8 – Publication MQTT**

L'ESP32 publie sur le broker MQTT sous les topics :

plante1/capteurs/sol plante1/capteurs/temperature plante1/  
capteurs/humidite\_air plante1/capteurs/luminosite plante1/system/  
etat plante1/system/last\_update

## **F9 – Dashboard utilisateur**

Le dashboard doit permettre :

- l'affichage temps réel,
- l'historique,
- les graphiques,
- les alertes.

# **7. Exigences techniques**

- Langage : C/C++ Arduino, framework Arduino sur ESP32
- Libraries : LiquidCrystal, Adafruit\_TSL2591, DHT, LoRa.h, PubSubClient
- Architecture : boucle non bloquante `millis()`
- Robustesse : gestion d'erreurs capteur, stabilité 24h

# **8. Critères d'acceptation**

## **Phase 1**

- Lecture automatique 60s OK
- Bouton → LCD 10s + lecture immédiate
- LED bleue synchronisée
- LEDs rouge/verte cohérentes
- Stabilité 24h

## **Phase 2**

- Trames LoRa reçues  $\geq 95\%$
- Publication MQTT correcte
- Dashboard fonctionnel

## **9. Livrables**

- Schéma Fritzing complet
- Code Arduino Mega
- Code ESP32 LoRa  $\rightarrow$  MQTT
- Documentation de câblage
- Cahier des charges (ce fichier)