

Dossier ressources : STATION METEO

1. Découverte du produit et de la problématique technique

Les Stations météo désignent des installations ou des dispositifs scientifiques dédiés à la mesure et à l'enregistrement des conditions météorologiques. Ces stations jouent un rôle crucial dans la collecte de données sur le climat et la météo, permettant des prévisions et des analyses précises. Les stations météo peuvent varier en taille et en complexité. Elles sont souvent équipées de divers instruments pour effectuer des mesures telles que :

- la température de l'air ;
- l'humidité relative ;
- la pression atmosphérique ;
- la vitesse et la direction du vent ;
- les précipitations ;
- la nébulosité.

Les stations météo jouent un rôle fondamental dans la surveillance du changement climatique. En collectant des données sur de longues périodes, elles aident à identifier des tendances telles que l'augmentation des températures mondiales, les modifications des schémas de précipitations et l'augmentation de la fréquence des événements météorologiques extrêmes. Les résultats obtenus permettent également de comprendre comment les activités humaines affectent le climat et de prendre des décisions éclairées sur la gestion des ressources naturelles. Dans des régions éloignées, les stations peuvent également fournir des données cruciales pour les prévisions de catastrophes naturelles, contribuant à la prévention et à la gestion des risques. Dans la station météo actuelle (Figure 2 et Figure 3), un automate mesure et affiche la vitesse et la direction du vent.

La problématique est de compléter ces mesures avec celle des précipitations. Ce domaine est la pluviométrie qui consiste en une mesure de la quantité de pluie tombée sur une région, généralement exprimée en millimètres. La mesure des précipitations présente de nombreux intérêts, notamment dans la gestion de la distribution des ressources en eau sur un territoire et permettent d'estimer les besoins d'irrigation en agriculture. Le pluviomètre est l'un des outils les plus couramment utilisés pour mesurer la quantité de pluie. Actuellement le pluviomètre utilisé (Figure 1) est composé d'un récipient gradué qui collecte l'eau et permet une lecture directe de la pluie tombée.



Figure 1 : pluviomètre à lecture directe



Supposons que notre pluviomètre indique 20 millimètres. Cela signifie que 20 litres d'eau sont tombés par mètre carré.

Si on calcule la quantité d'eau tombée sur un hectare :

- 20 mm correspond à $20 \text{ L} \cdot \text{m}^{-2}$;
- or 1 hectare = $10\,000 \text{ m}^2$;
- soit $20 \times 10\,000 = 200\,000 \text{ L}$;
- donc 200 000 litres d'eau sont tombés sur 1 ha, soient 200 m^3 .

Si la lecture directe est possible, elle ne permet pas de rendre le système autonome. Il n'est pas tenu compte de la durée, de plus l'affichage et le transfert des informations sont alors impossibles. La problématique est donc de modifier la structure et donc de remplacer le pluviomètre à lecture directe par un capteur ad hoc.

a) Caractéristiques de la station météo actuelle :

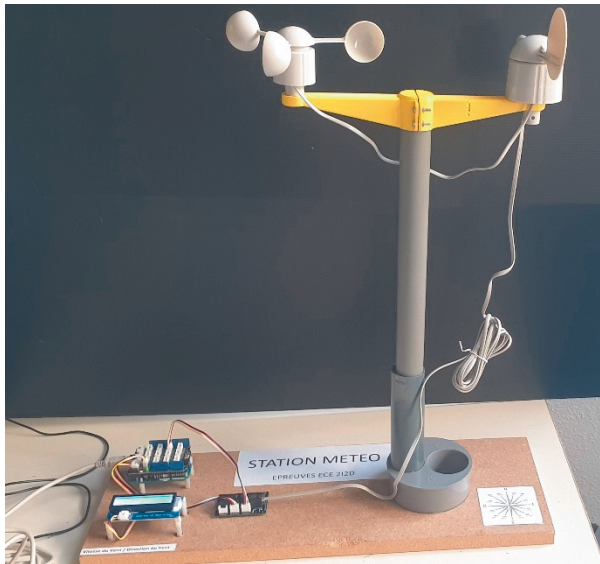


Figure 2 : station météo actuelle

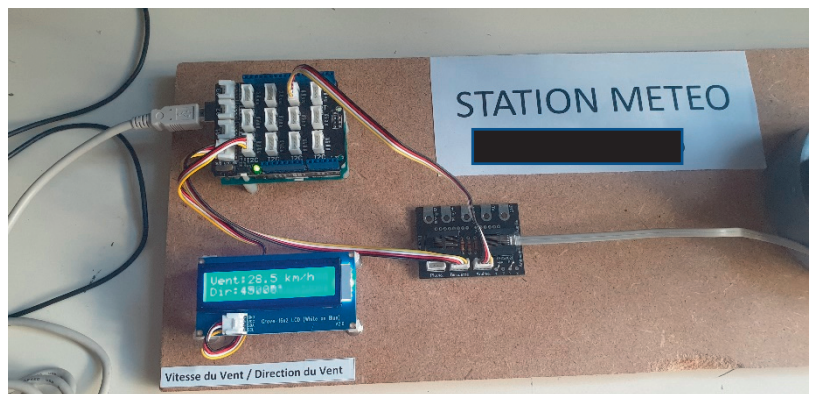


Figure 3 : cartes électroniques de la Station

- l'anémomètre intègre un petit aimant qui en passant devant un contact (ILS) ferme brièvement celui-ci lorsque les coupelles de l'anémomètre tournent (une vitesse de vent d'environ $2,4 \text{ km} \cdot \text{h}^{-1}$ génère la fermeture impulsionnelle du contact par seconde) ;
- la girouette génère une variation de la valeur de sa résistance en fonction de son orientation. Cette variation n'est pas proportionnelle à la direction (la girouette délivre 16 valeurs de résistances différentes en fonction de sa position - cette dernière est donc uniquement capable de donner la direction du vent selon 16 positions) ;
- le pluviomètre à lecture directe (Figure 1) se présente sous forme d'un entonnoir gradué en plastique transparent indiquant la quantité d'eau recueillie pour connaître la quantité

tombée par mètre carré. Après chaque mesure, opération à effectuer régulièrement en cas de fortes précipitations, le pluviomètre doit être vidé manuellement. La mesure est ensuite inscrite dans un registre.

b) Diagramme des exigences

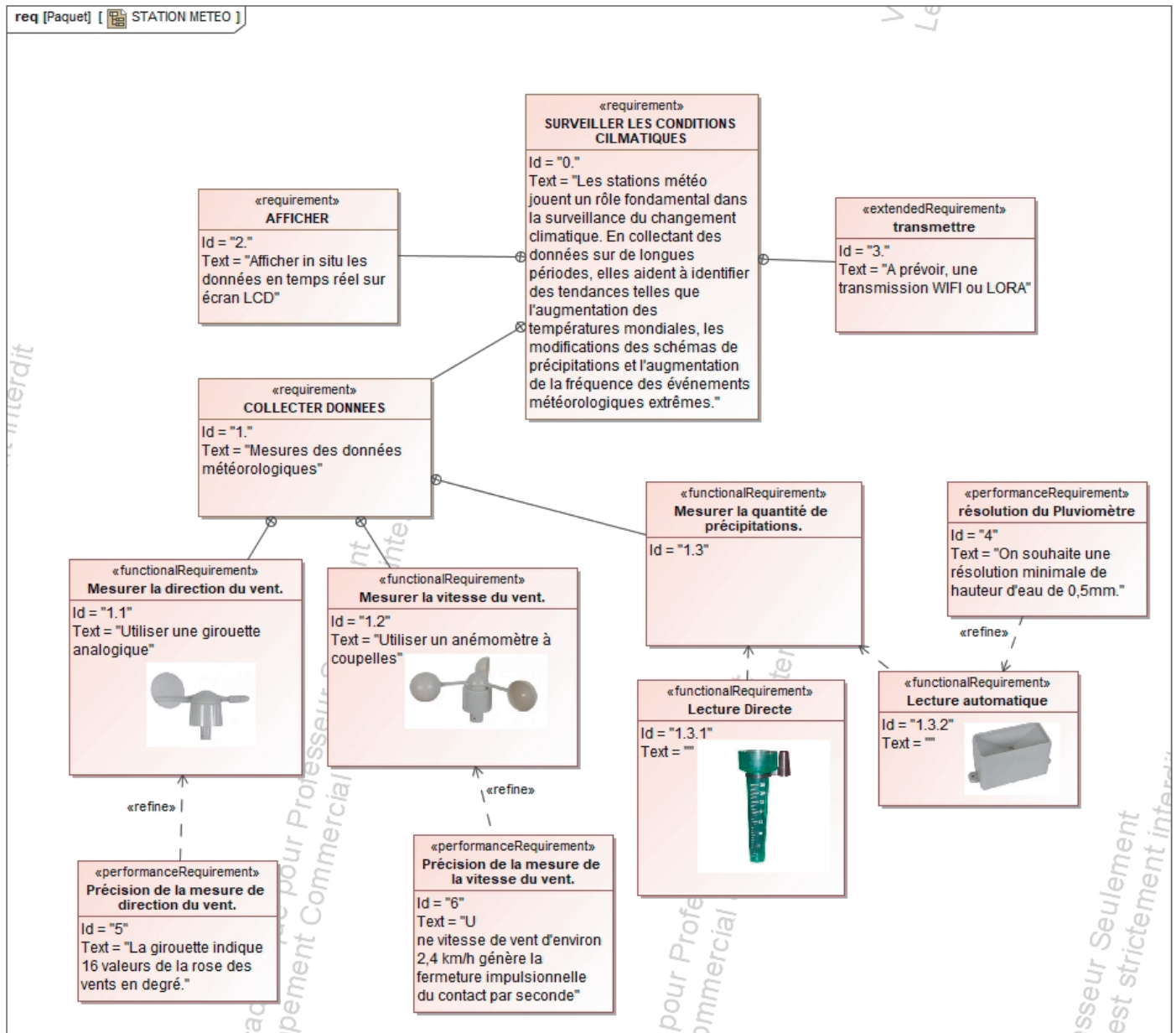


Figure 4 : SYSML diagrammes des exigences

2. Conception

a) Justification du capteur

Les pluviomètres à augets basculants qui sont plus précis et automatisés. Ces instruments sont équipés de petits récipients capteurs (augets ou godets) qui basculent après avoir recueilli une certaine quantité de précipitations. Un enregistreur électronique enregistre le nombre de basculements et calcule ainsi la quantité de pluie tombée. Ces pluviomètres sont souvent utilisés dans les stations météo professionnelles.



Figure 5 : pluviomètre à augets basculants avec le bloc ouvert

Après chaque remplissage, le godet se vide automatiquement afin de pouvoir être prêt à se remplir à nouveau. L'information du capteur est accessible via une prise RJ11 déportée sur un câble de 2.5 m.

Principe d'utilisation du pluviomètre : L'exploitation du capteur est la suivante : il faut lui associer une résistance de tirage vers le haut ou pull up (Figure 6) et câbler les sorties du capteur entre cette résistance de tirage et la masse. Le microcontrôleur devra simplement détecter les impulsions passagères (niveau logique bas) qui lui indiqueront que 0.2794 mm d'hauteur d'eau sont tombées (à chaque impulsion détectée).

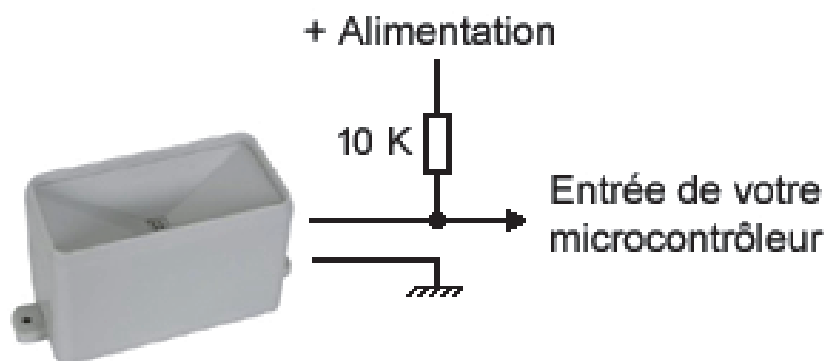
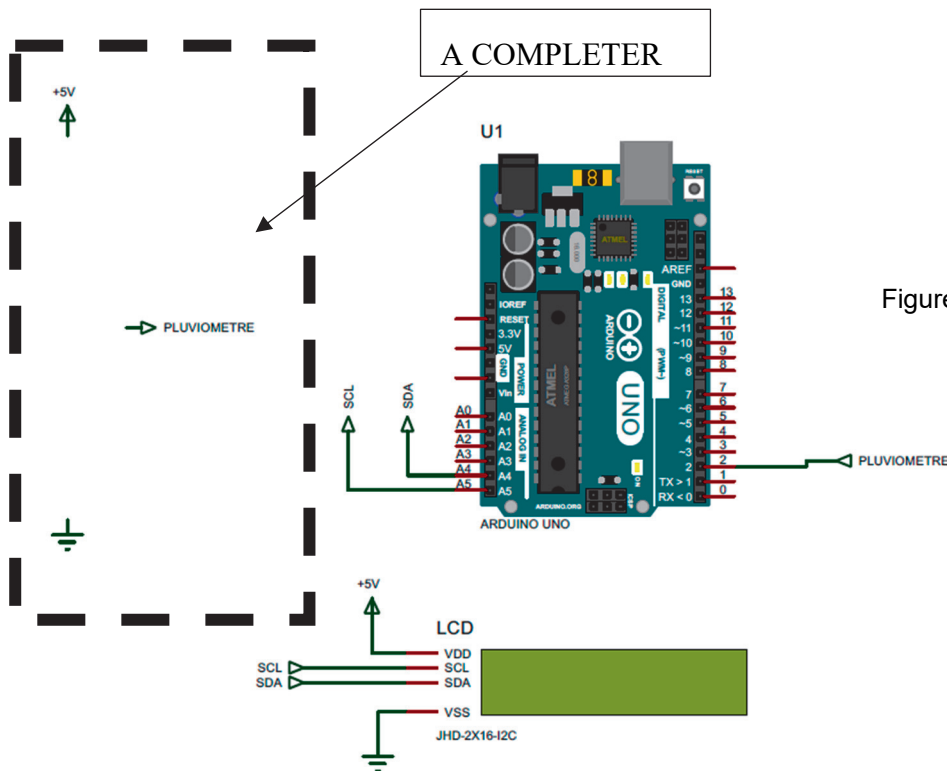


Figure 6 : câblage du capteur avec résistance de pull up

b) Schéma structurel partiel à compléter



c) Principe de l'interruption sous Arduino :

Les interruptions permettent au microcontrôleur d'exécuter une fonction lorsqu'un évènement survient sur une des broches d'interruption. Plutôt que de lire constamment la valeur d'un capteur, le programme ne va se déclencher que lorsque la valeur du capteur va changer. La syntaxe pour initialiser une interruption est la suivante:

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)
```

Avec :

- pin, numéro de la broche utilisée. Sur la carte Arduino UNO, il y a deux broches qui supportent les interruptions : les broches 2 et 3. La 3 sera utilisée pour le pluviomètre ;
- ISR, la fonction à activer lorsque l'évènement est détecté. Dans le programme la fonction se nomme interruptPluviometre ;
- mode est le mode de détection utilisé et choisi parmi :
 - **LOW** pour activer la détection lorsque la broche est à l'état BAS ;
 - **CHANGE** pour activer la détection lorsque l'état de la broche change ;
 - **RISING** pour activer la détection lorsque l'état de la broche passe de BAS à HAUT ;
 - **FALLING** pour activer la détection lorsque l'état de la broche passe de HAUT à BAS.

d) Programme à compléter : **pluviometre__eleve.ino**

```
#include "rgb_lcd.h"
rgb_lcd lcd;
const int colorR = 255;
const int colorG = 0;
const int colorB = 0;

const int pinPluviometre = .....; // compléter le numéro de broche
choisi pour le pluviomètre

volatile int comptage = 0;           // Compteur de basculements
unsigned long dernierTemps = 0;      // Pour l'anti-rebond
const unsigned long delaiAntiRebond = 500; // 500 ms de délai anti-
rebond

#define VALEUR_PLUVIOMETRE ..... //compléter valeur en mm d'eau à
chaque bascule d'auget.

float pluvio10(0); //pluie sur 10 secondes.
float cumul_pluie(0); // cumul de pluie.
unsigned long previousMillis= 0;
unsigned long previousMillis2= 0;
unsigned long delaiProgramme = 10000;
//variable représentant les 10 sec d'acquisition de comptage de
basculement de godet.
void setup(){
// Il nous faut déclarer l'interruption. Le premier paramètre est le
numéro de l'interruption.
// Le deuxième sera le nom de la fonction chargée de traiter
l'interruption.
// Le troisième paramètre est le choix du front de déclenchement.

attachInterrupt(digitalPinToInterrupt(.....),interruptPluviometre,.....
.....);

// Compléter les 2 informations manquantes
```



```
// Initialisation de l'afficheur
lcd.begin(16, 2);
lcd.setRGB(colorR, colorG, colorB);

lcd.print("STATION METEO");
delay(1000);
lcd.clear();
}
void loop() {
    // millis() contient le temps écoulé depuis le lancement du
    // programme.
    // Affiche le cumul de précipitations toutes les 5 secondes
    static unsigned long dernierAffichage = 0;
    if (millis() - dernierAffichage >= 5000) {
        dernierAffichage = millis();
        float mmPluie = comptage * VALEUR_PLUIVIOMETRE; //1 basculement

        //Affichage sur moniteur série
        Serial.print("Pluie accumulée : ");
        Serial.print(mmPluie, 2);
        Serial.println(" mm");

        //Affichage sur l'écran 2 lignes
        lcd.setCursor(0, 0);
        lcd.print("Pluie en mm :");
        lcd.setCursor(0, 1);
        lcd.print(mmPluie);
        lcd.print(" mm.");
    }
}

// Fonction appelée à chaque front descendant sur l'entrée
// (basculement ILS)
void interruptionPluie() {
    unsigned long maintenant = millis();
    if (maintenant - dernierTemps > delaiAntiRebond) {
        comptage++;
        dernierTemps = maintenant;
    }
}
```

3- Simulation

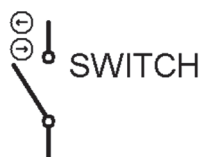


Figure 8 : modèle utilisé pour la simulation du pluviomètre (Switch)

Sous Proteus /ISIS, ouvrir le fichier : pluviometre.pdsprj

L'intégration du code source de la carte de développement Arduino sous ISIS se fait en double-cliquant sur la carte puis en cliquant sur l'icône de sélection de fichier (Figure 9). Il ne reste ensuite qu'à sélectionner le fichier HEX généré sous l'IDE d'Arduino.

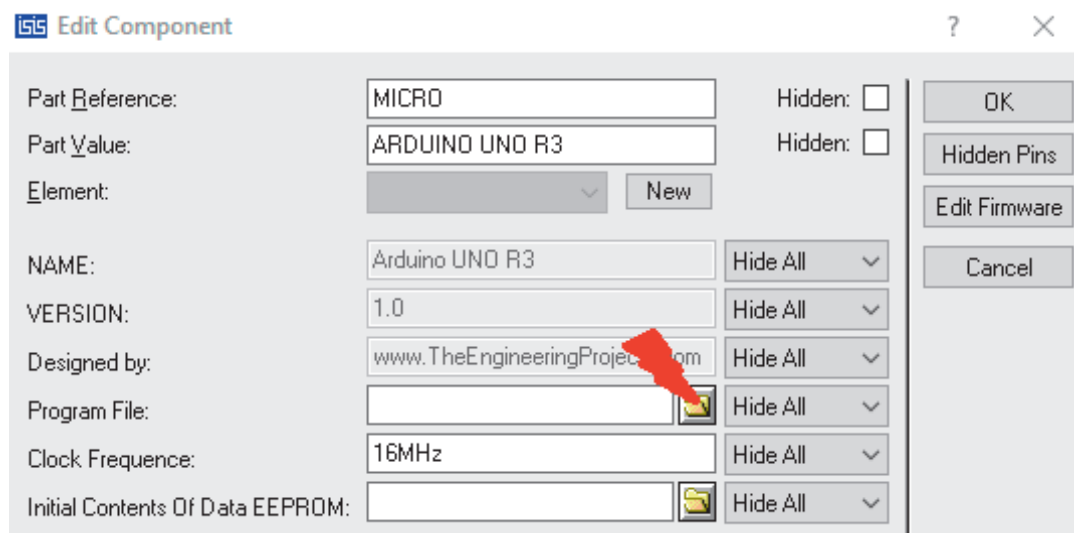


Figure 9 : intégration du code sous Protéus

4- Expérimentation

Pour effectuer l'expérimentation, le matériel suivant est à disposition :

- une carte de développement Arduino ;
- un shield Grove de connexion ;
- un capteur Pluviomètre à godet pour Arduino ;
- une carte interface pour Capteurs ;
- un afficheur LCD 2*16 ;
- des câbles de connexion.




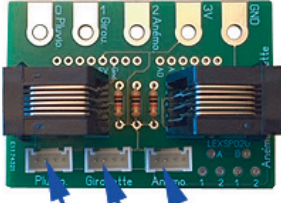
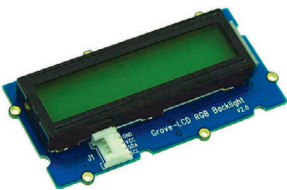
 <p>ARDUINO UNO REV3</p>	 <p>GROVE BASE SHIELD ARDUINO</p>
 <p>CAPTEUR PLUVIOMETRE A GODET LEXCA001</p>	 <p>Attention au sens des connecteurs</p> <p>INTERFACE DEV-LEX2 (pour Grove) DES CAPTEURS METEO</p>
 <p>AFFICHEUR LCD 2*16 Grove</p>	

Figure 10 : liste du matériel à disposition