

CAHIER DES CHARGES

STATION

MÉTÉO

BASIL DIEUX
MATTHIAS LUTZ
MANON CHABANNES
MARION DABOUT

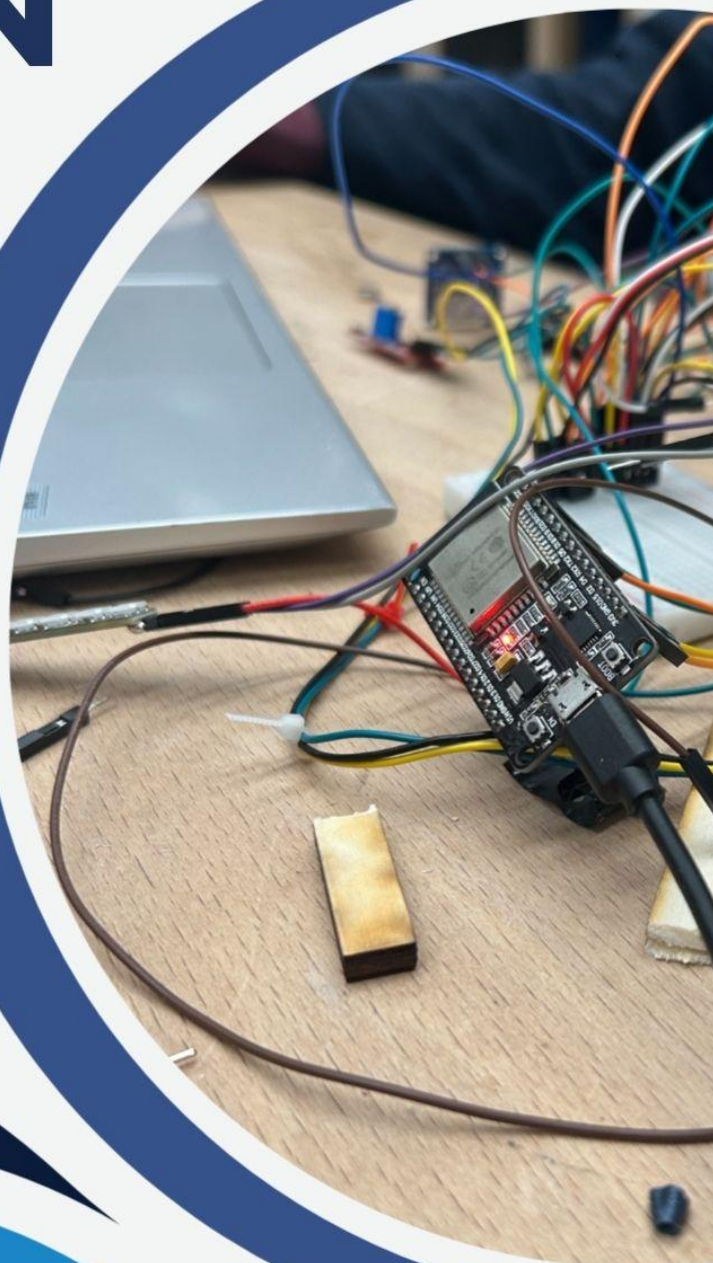


Table des matières

Sujet	4
Contexte	4
Identification des problèmes dans le contexte	5
Proposition de réponse aux problématiques avec l'objet connecté.....	5
Description des capteurs et actionneurs utilisés.....	6
Répartition des tâches au sein du groupe	6
Mise en place des outils pour travailler en groupe	7
Présentation de l'architecture logicielle du projet sous forme de logigramme	7
Schéma de câblage (Fritzing, Autodesk Tinkercad, KiCAD...)	8
Base de données	9
Comment NodeRed est il intégré dans votre projet ?	10
Rôle de Node-Red dans l'architecture du projet.....	10
Architecture entre Node-Red, MQTT et Arduino	10
Flow 1 – Connexion au broker MQTT.....	10
Flow 2 – Traitement des données et calculs.....	11
Flow 3 – Automatisations et alertes.....	11
Dashboard Node-Red	11
Contribution de Node-Red dans le projet	12
Plans de la maquette ou des modèles 3D nécessaires pour le projet	12

Sujet

Conception et réalisation d'une **Station Météo connectée** capable de mesurer plusieurs paramètres environnementaux, d'afficher les données localement et de les envoyer à un serveur Node-RED via un protocole MQTT, avec la possibilité de déclencher des actions selon les conditions mesurées.

Contexte

Dans un contexte où le suivi de l'environnement et les bâtiments intelligents prennent de plus en plus d'importance, une station météo connectée permet de récupérer en temps réel différentes informations sur les conditions extérieures. Ces données peuvent servir aussi bien à optimiser la consommation énergétique d'un bâtiment, automatiser certains équipements, ou tout simplement suivre la météo à une localisation précise.

Identification des problèmes dans le contexte

La mise en place d'une station météo extérieure implique plusieurs défis techniques majeurs. Le premier concerne la **protection de l'électronique contre les intempéries**, car l'ensemble doit résister à la pluie, à l'humidité et aux infiltrations tout en évitant la surchauffe dans un boîtier exposé au soleil. Certains capteurs, comme ceux de **luminosité ou de vent**, doivent rester à l'extérieur du boîtier pour fonctionner correctement, ce qui pose **des contraintes de fixation, d'étanchéité et de résistance mécanique**, notamment face aux rafales. La **fiabilité des mesures** représente également un enjeu important : **la température peut être faussée par la chaleur interne de l'électronique, le capteur à effet Hall peut produire des impulsions erronées si le rotor vibre, et certains capteurs sensibles, comme ceux de qualité de l'air**, nécessitent une stabilisation et une calibration précise. Enfin, la station doit maintenir une **connexion Wi-Fi stable** pour transmettre ses données via MQTT, malgré un emplacement extérieur **parfois éloigné du routeur**.

Proposition de réponse aux problématiques avec l'objet connecté

Pour répondre aux différentes problématiques identifiées, notre station météo connectée apportera des solutions adaptées aussi bien aux contraintes extérieures qu'aux besoins de fiabilité. L'électronique sera protégée dans un **boîtier résistant aux intempéries**, tandis que les capteurs nécessitant une exposition directe seront placés à l'extérieur de manière sécurisée pour garantir des mesures cohérentes. La précision des données sera améliorée grâce à une **calibration adaptée** et un traitement logiciel permettant de corriger les variations brusques ou les valeurs aberrantes. La communication entre l'Arduino et le serveur sera assurée par MQTT, ce qui garantit une **transmission stable et continue** même en cas de conditions variables. Enfin, l'ajout d'un tableau de bord et de quelques automatisations permettra de **surveiller les données en temps réel** et de rendre la station plus réactive et plus fiable.

Description des capteurs et actionneurs utilisés

Capteurs

- Température/humidité/Pression atmosphérique → BME280
- Luminosité → Luxmètre (SEN0097)
- Vent → Un capteur à effet Hall, pour créer un Anémomètre (Module à effet hall KY024LM)
- Qualité de l'air (SENCCS811V1)

Actionneurs

- Bandeau LED NeoPixel (ADA1426) → Affiche les alertes météo avec un code couleur (bleu : froid, rouge : chaud, jaune : beaucoup de vent)
- Buzzer pour les alertes (alerte météo, vent fort, inondation, ...)
- Écran LCD 16x2 (I2C)

Répartition des tâches au sein du groupe

Rôle	Membre	Responsabilités
Conception hardware et câblage => Un Préposé à la soudure : formation au Crunch	Basile Dieux	Soudures, câblages, fixation capteurs/actionneurs
Conception software pour s'interfacer avec les capteurs et actionneurs	CHABANNES Manon	Code Arduino : lecture des capteurs, commande actionneurs
Conception software pour la communication MQTT via wifi.	Marion DABOUT	Implémentations des topics MQTT, échanges bidirectionnels
Mise en place de la solution serveur NodeRed	Matthias LUTZ	Création du Dashboard, stockage des données, automatisations, Base de données

Mise en place des outils pour travailler en groupe

- Communication via Teams
- OneDrive → création d'un dossier partager (Docs, Word...) avec un document Word dans lequel nous nous partageons précisément les différentes tâches.
- Points d'avancement à la fin de certains cours d'IF3B

Présentation de l'architecture logicielle du projet sous forme de logigramme

Voici un logigramme de base pour faire marcher la station météo

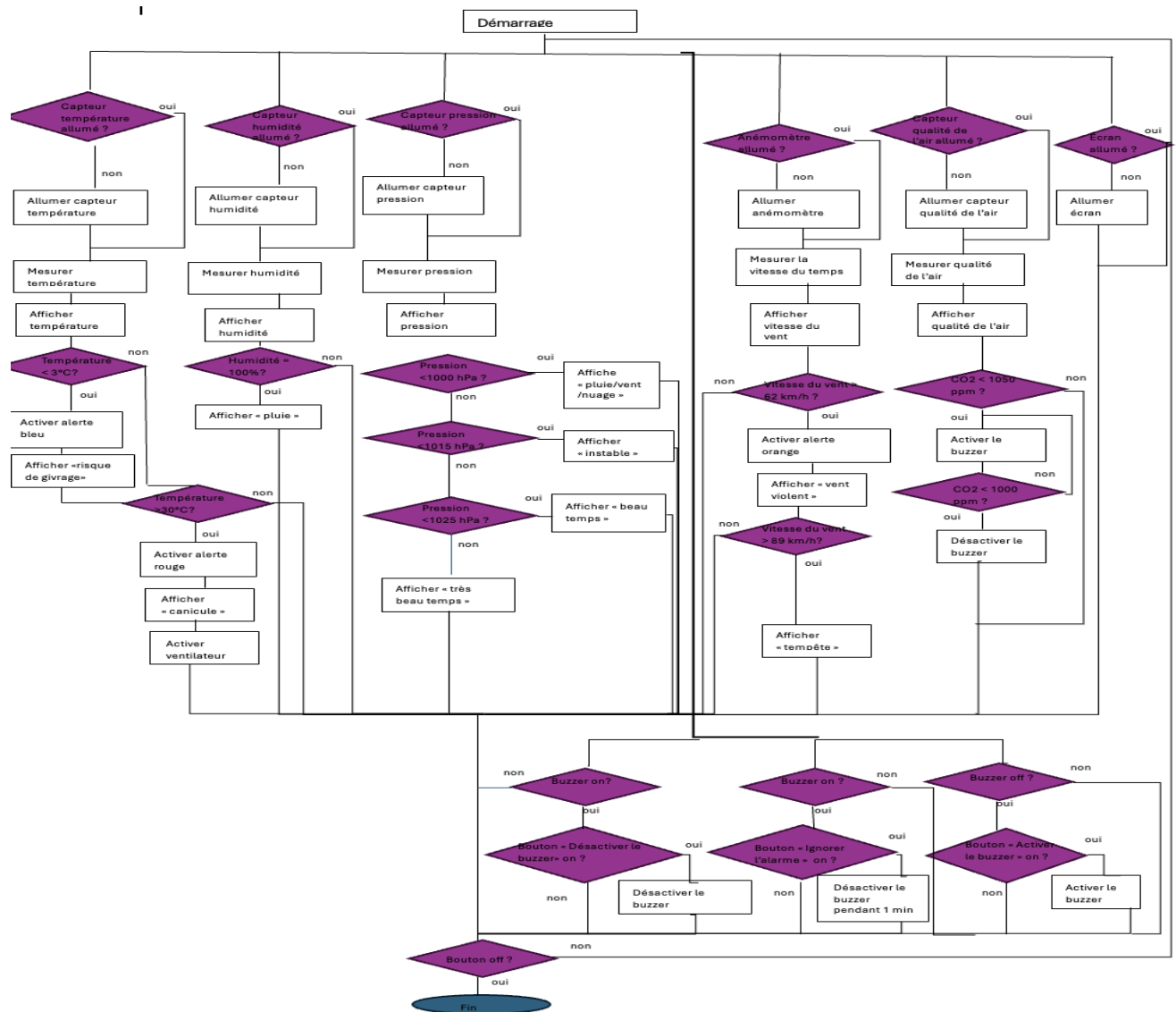
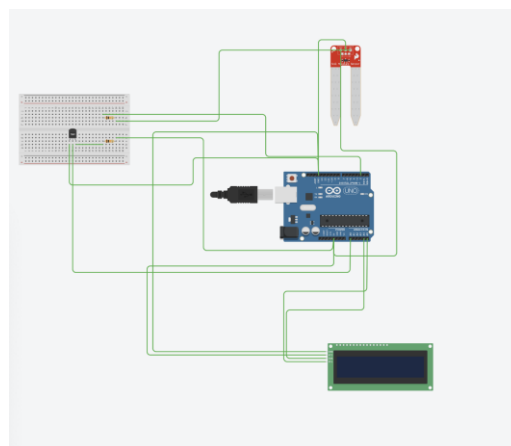
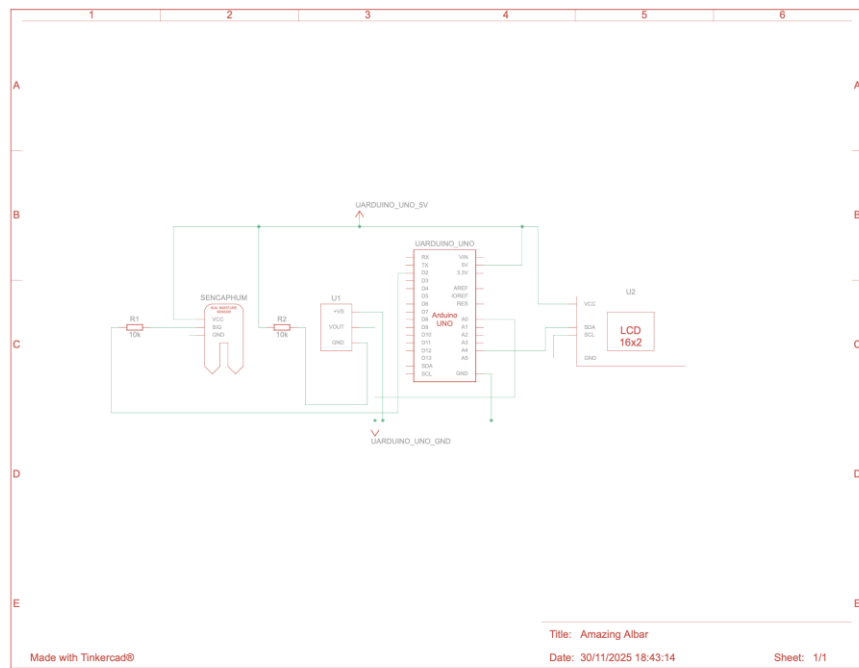


Schéma de câblage (Fritzing, Autodesk Tinkercad, KiCAD...)





Base de données

CREATE TABLE mesures (

id INTEGER PRIMARY KEY AUTOINCREMENT,

timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,

temperature REAL,

humidite REAL,

pression REAL,

luminosite REAL,

vent_m_s REAL

);

Exemple de table simple que l'on pourrait utiliser sur MySQL. Elle stockerait les différentes mesures au cours du temps, et permettrait également de créer des statistiques (des graphiques, des tableaux, ...) pour rendre l'analyse des données plus visible.

Comment NodeRed est il intégré dans votre projet ?

➤ **Rôle de Node-Red dans l'architecture du projet**

Node-RED est un outil de programmation visuelle qui permet de créer des traitements et des interfaces en connectant des blocs, appelés *nœuds*. Dans notre station météo, Node-RED joue le rôle de serveur central. C'est lui qui récupère les données envoyées par l'Arduino via MQTT, les organise, les transforme, puis les affiche à l'utilisateur sur un tableau de bord. Il permet aussi d'envoyer des commandes vers l'Arduino, ce qui garantit la communication bidirectionnelle imposée dans le cadre du projet.

➤ **Architecture entre Node-Red, MQTT et Arduino**

La communication entre Node-RED et notre carte Arduino se fait grâce au protocole MQTT. L'Arduino publie régulièrement ses mesures (température, humidité, pression, luminosité, fréquence du vent, etc.) sur différents *topics* tels que *meteo/temperature* ou *meteo/vent/freq*. Node-RED s'abonne à ces topics pour recevoir les valeurs en temps réel. À l'inverse, Node-RED peut publier sur des topics de commande comme *meteo/cmd/ventilateur* pour envoyer une instruction à l'Arduino.

➤ **Flow 1 – Connexion au broker MQTT**

Le premier flow de Node-RED s'occupe uniquement de la communication avec le broker MQTT (Mosquitto). Il contient tous les nœuds MQTT nécessaires pour recevoir les mesures envoyées par l'Arduino et pour transmettre des commandes vers celui-ci. Les nœuds MQTT-IN sont configurés pour écouter des topics comme *meteo/temperature*, *meteo/humidite* ou *meteo/vent/freq*. De l'autre côté, les nœuds MQTT-OUT permettent d'envoyer des instructions telles que l'allumage d'une LED, l'envoi d'un signal, ...

➤ **Flow 2 – Traitement des données et calculs**

Le deuxième flow est dédié au traitement des données avant leur affichage. Les valeurs reçues (température, humidité, pression, luminosité) sont d'abord restructurées en JSON, filtrées pour retirer les valeurs aberrantes, puis converties dans les unités les plus adaptées à l'affichage. Node-RED prépare ensuite ces données pour les widgets du Dashboard, comme les jauges ou les graphiques.

Une partie spécifique de ce flow concerne la vitesse du vent. Notre anémomètre, basé sur un capteur à effet Hall, génère des impulsions à chaque tour. L'Arduino calcule la fréquence de ces impulsions et la transmet à Node-RED via le topic meteo/vent/freq. Dans un nœud Function, Node-RED applique la formule $v = K \times f$, où f est la fréquence et K un coefficient de calibration déterminé expérimentalement. Cela permet d'obtenir précisément la vitesse du vent en m/s et en km/h.

➤ **Flow 3 – Automatisations et alertes**

Le troisième flow gère tous les comportements automatiques du système. En fonction des valeurs reçues, Node-RED peut déclencher certaines actions : par exemple activer un ventilateur si la température dépasse un seuil fixé, afficher une alerte si le vent devient trop fort ou encore allumer automatiquement un éclairage lorsque la luminosité baisse trop. Node-RED permet aussi de détecter l'absence de données (timeout MQTT) et de prévenir l'utilisateur via le Dashboard. Les nœuds utilisés pour ces automatisations sont principalement les nœuds Switch, Change, Function, Delay et MQTT-OUT.

➤ **Dashboard Node-Red**

Le Dashboard constitue l'interface directe entre le système et l'utilisateur. Il permet de visualiser en temps réel les données de la station météo : température, humidité, pression, luminosité et vitesse du vent. Chaque mesure peut être affichée sous forme de jauge, de barres ou de graphiques historiques. L'utilisateur dispose également de plusieurs commandes, comme un bouton pour activer ou désactiver le ventilateur, un réglage du seuil d'alerte de température, un interrupteur pour passer en mode automatique ou manuel, ou encore un bouton pour demander une mise à jour des données. Il doit permettre l'utilisation sur PC et éventuellement sur smartphones.

➤ ***Contribution de Node-Red dans le projet***

Node-RED occupe une place essentielle dans notre architecture globale. C'est lui qui centralise toutes les données captées par l'Arduino, qui les traite et qui les affiche. Il réalise aussi certains calculs importants, comme la conversion de la fréquence du capteur Hall en vitesse du vent. Grâce à ses capacités d'automatisation, Node-RED peut prendre des décisions et déclencher des actions sans intervention humaine. Enfin, il sert d'interface utilisateur, ce qui permet de surveiller l'état du système en temps réel et de contrôler manuellement certains actionneurs. En résumé, Node-RED assure à la fois le rôle de serveur, d'outil de traitement et d'interface de supervision.

Plans de la maquette ou des modèles 3D nécessaires pour le projet

Boîte maquette sur base box

BOÎTE FERMÉE

Boîte complètement fermée

► Réglages pour les entures crénelées

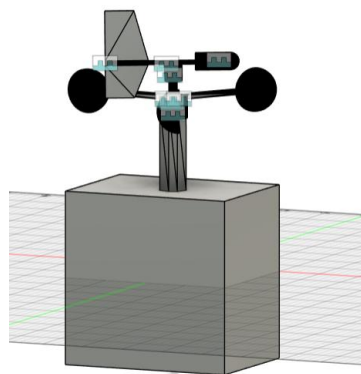
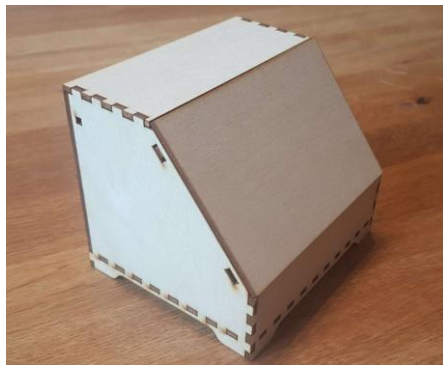
▼ Réglages de Boîte Fermée

x largeur intérieure en mm (si on ne choisit pas « extérieur »)
y profondeur intérieure en mm (si on ne choisit pas « extérieur »)
h hauteur intérieure en mm (si on ne choisit pas « extérieur »)
extérieur ☒ treat sizes as outside measurements ⓘ

▼ Réglages par défaut

épaisseur thickness of the material (in mm) ⓘ
format format of resulting file ⓘ
brides width of tabs holding the parts in place (in mm)(not supported everywhere) ⓘ
qr_code ☐ Add a QR Code with link or command line to the generated output
debug ☐ print surrounding boxes for some structures ⓘ
labels ☒ label the parts (where available)
référence print reference rectangle with given length (in mm)(zero to disable) ⓘ
inner_corners style for inner corners ⓘ
brûlage burn correction (in mm)(bigger values for tighter fit) ⓘ
spacing spacing around parts (multiples of thickness [: extra space in mm]) ⓘ

[Générer](#) [Download](#) [Save to URL](#) [QR Code](#)



Nous avons donc opté pour la confection d'une boîte (réalisé à la découpe laser) qui va donc permettre de ranger et cacher la carte et différents composants. Nous rajouterons au-dessus de celle-ci l'aéromètre permettant, à l'aide d'un capteur à

effet hall, de déduire la vitesse du vent. Nous rajouterons par la suite sur cette boîte quelques ouvertures nécessaires au bon fonctionnement de certains capteurs comme le capteur de luminosité.

