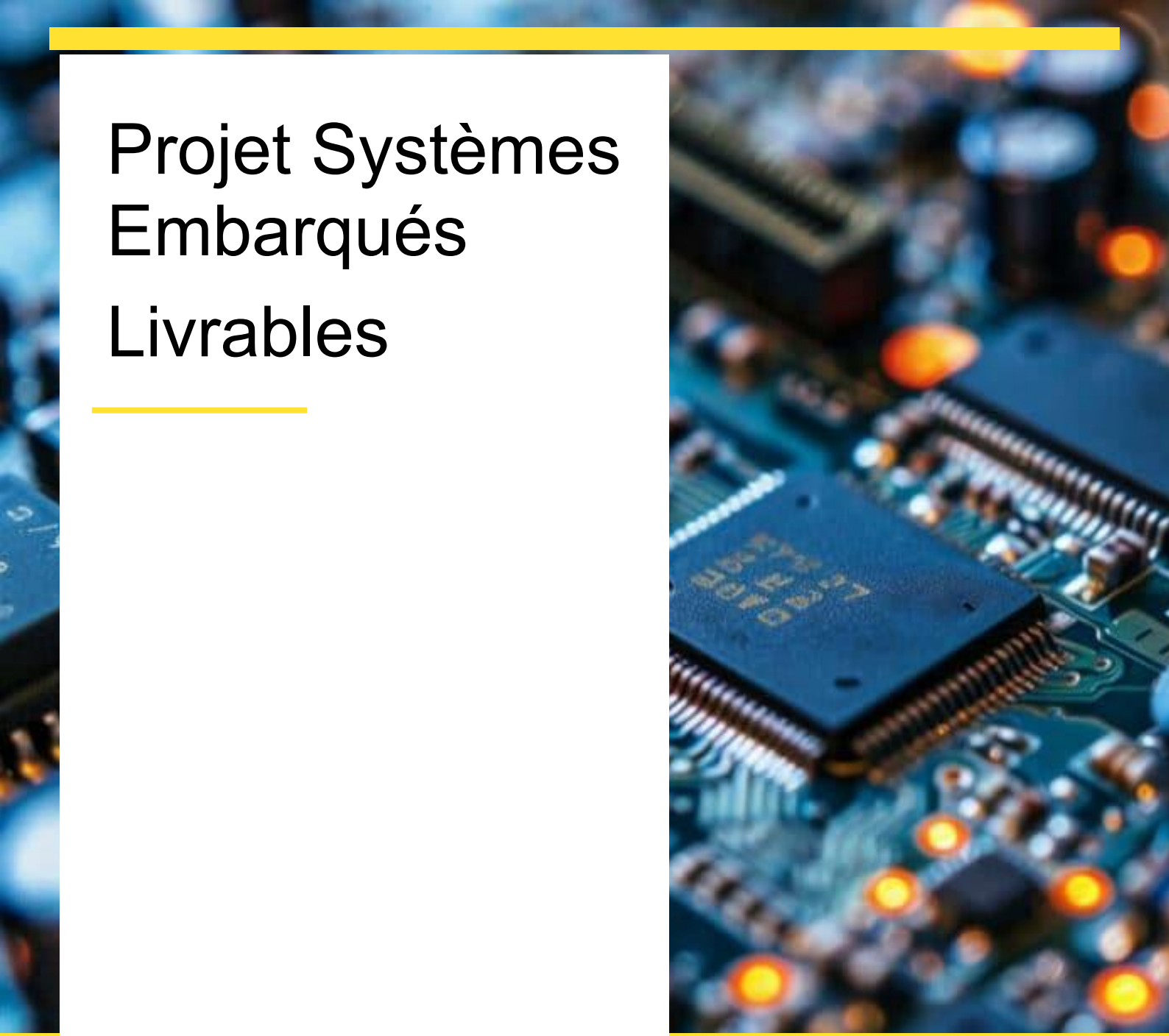


Projet Systèmes Embarqués

Livrables

24 OCTOBRE 2025

Chef de Groupe : Paul Calais
Membre : Lucas Baye
Enzo Verdin

CESI
ÉCOLE D'INGÉNIEURS

Table des matières

Contexte :

L'Agence Internationale pour la Vigilance Météorologique (AIVM) souhaite déployer des navires équipés de stations météo embarquées capables de mesurer différents paramètres environnementaux (température, pression, humidité, luminosité, position GPS) afin d'améliorer la détection et la prévention des phénomènes météorologiques dangereux.
Un prototype sera développé à l'aide d'une carte Arduino (ATmega328) et de plusieurs capteurs. Le système devra être utilisable un membre d'équipage non spécialiste.

Problématique :

Comment concevoir et modéliser une station météo embarquée capable de collecter, enregistrer et signaler des données météorologiques dans un navire en mer ?

Contraintes :

-Matériel imposé → Microcontrôleur ATmega328
-interfaces matérielles → bus de communication standards (I2C / SPI /UART/Entrées analogiques)
-gestion du stockage
-neuf capteurs
-contrainte d'environnement
-contrainte de facilité d'utilisation

Livrable 1 :

Objectif :

Présenter sous formes de diagrammes/schémas/algorithmes les différentes fonctionnalités de la station météo du projet.

Diagramme UML/SYSML :

Définitions :

SysML (Systems Modeling Language) est un langage de modélisation. Il permet la spécification, l'analyse, la conception, la vérification et la validation de nombreux systèmes.

UML est un moyen de visualiser des systèmes et des logiciels à l'aide du langage de modélisation unifié. Les ingénieurs logiciels créent des diagrammes UML pour comprendre les conceptions, l'architecture du code et la mise en œuvre proposée de systèmes logiciels complexes

Nous allons présenter plusieurs diagrammes permettant de mieux comprendre notre projet :

-Le diagramme de cas d'utilisation, qui décrit les interactions entre les acteurs et le système, et met en évidence les principales fonctionnalités attendues.
-Le diagramme de séquence, qui montre la chronologie des échanges entre les différents éléments du système.
-Le diagramme d'activité, qui représente le déroulement logique des actions et processus internes.

Ces diagrammes constituent des outils pour structurer la conception et assurer la cohérence pour notre projet

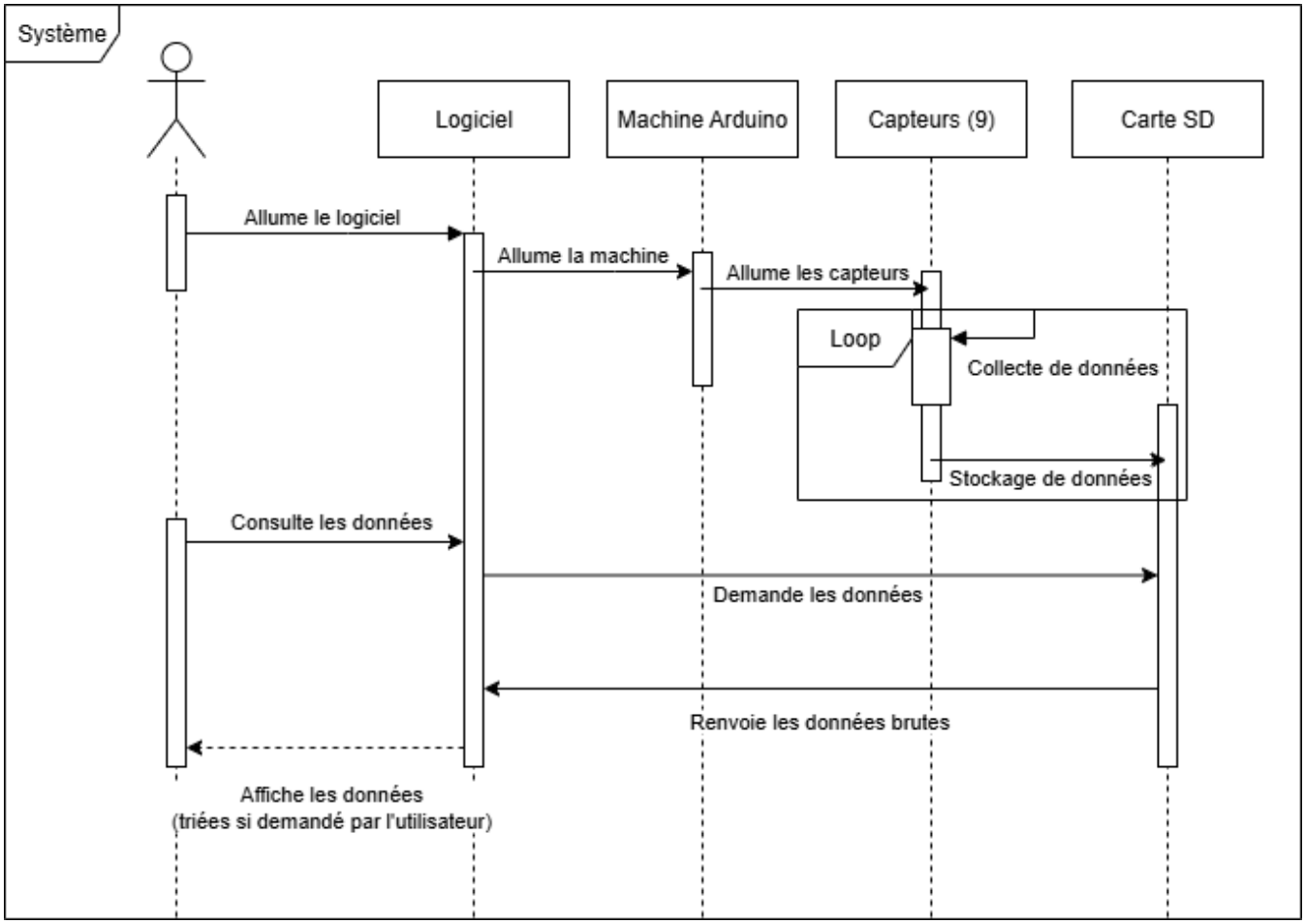
Dans cette première phase du projet, les diagrammes réalisés (cas d'utilisation, séquence et activité) s'organise sur le fonctionnement principal du système en mode standard. Nous l'avons fait pour se focaliser sur le système général de la station météo.

- L'initialisation des capteurs et des modules,
- La collecte et la sauvegarde des données environnementales,

- La signalisation de l'état du système via la LED RGB,
- Et l'interaction de base avec l'utilisateur (consultation et enregistrement).

Les modes configuration, maintenance et économique n'ont pas été intégrés dans les diagrammes de cas d'utilisation et de séquence, car ils correspondent à des fonctionnalités avancées.

Diagramme de séquence :



Définitions :

Logiciel : interface utilisateur permettant d'interagir avec la machine Arduino.

Machine Arduino : microcontrôleur qui contrôle les capteurs et gère la communication avec la carte SD.

Capteurs (9) : dispositifs qui collectent les données physiques (pression atmosphérique, température de l'air et de l'eau, hygrométrie, GPS, luminosité, force du courant marin et du vent et taux de particule fines.)

Carte SD : mémoire externe servant à stocker les données collectées.

Etape :

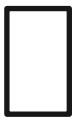
1) Le système (utilisateur externe) démarre le logiciel

2) Le logiciel envoie une commande pour activer la carte Arduino.
3) L'Arduino active les capteurs connectés.
4) Les capteurs mesurent et transmettent les données à l'Arduino.
5) L 'Arduino enregistre ces données sur la carte SD.
6) Le logiciel demande à l'Arduino les données enregistrées.
7) L'Arduino renvoie les données collectées brutes.
8) Le logiciel affiche les données à l'utilisateur.

Légende :

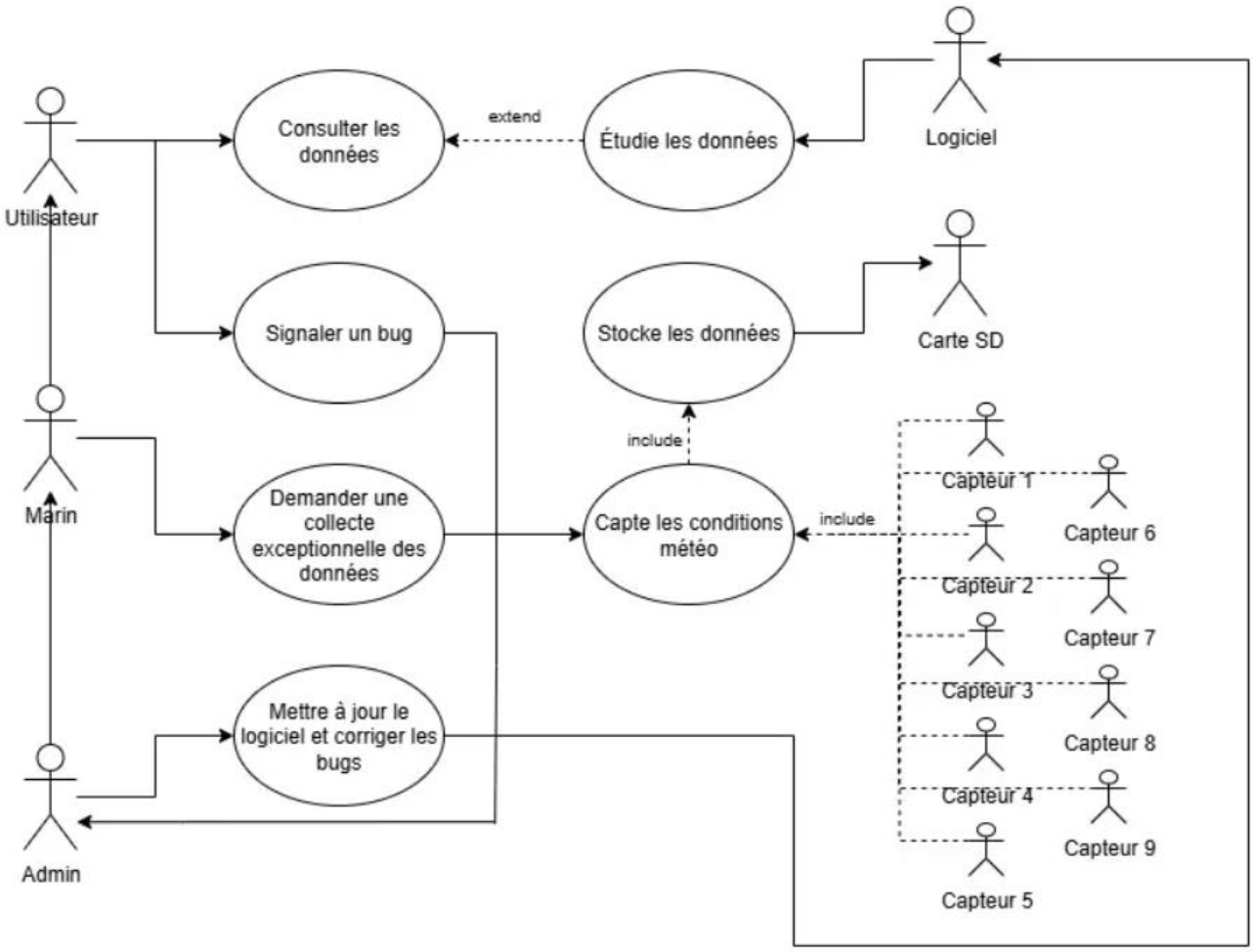
Loop = Le bloc "Loop" indique une répétition (il représente la collecte continue des données par les capteurs et leur stockage sur la carte SD.)

Ligne de vie d'un objet



Représente les activations, ce sont les périodes pendant lesquelles un objet exécute une action

Diagramme de cas d'utilisation :



1. Acteur :

1)Utilisateur

2)Marin

3)Admin (Administrateur)

Légende :

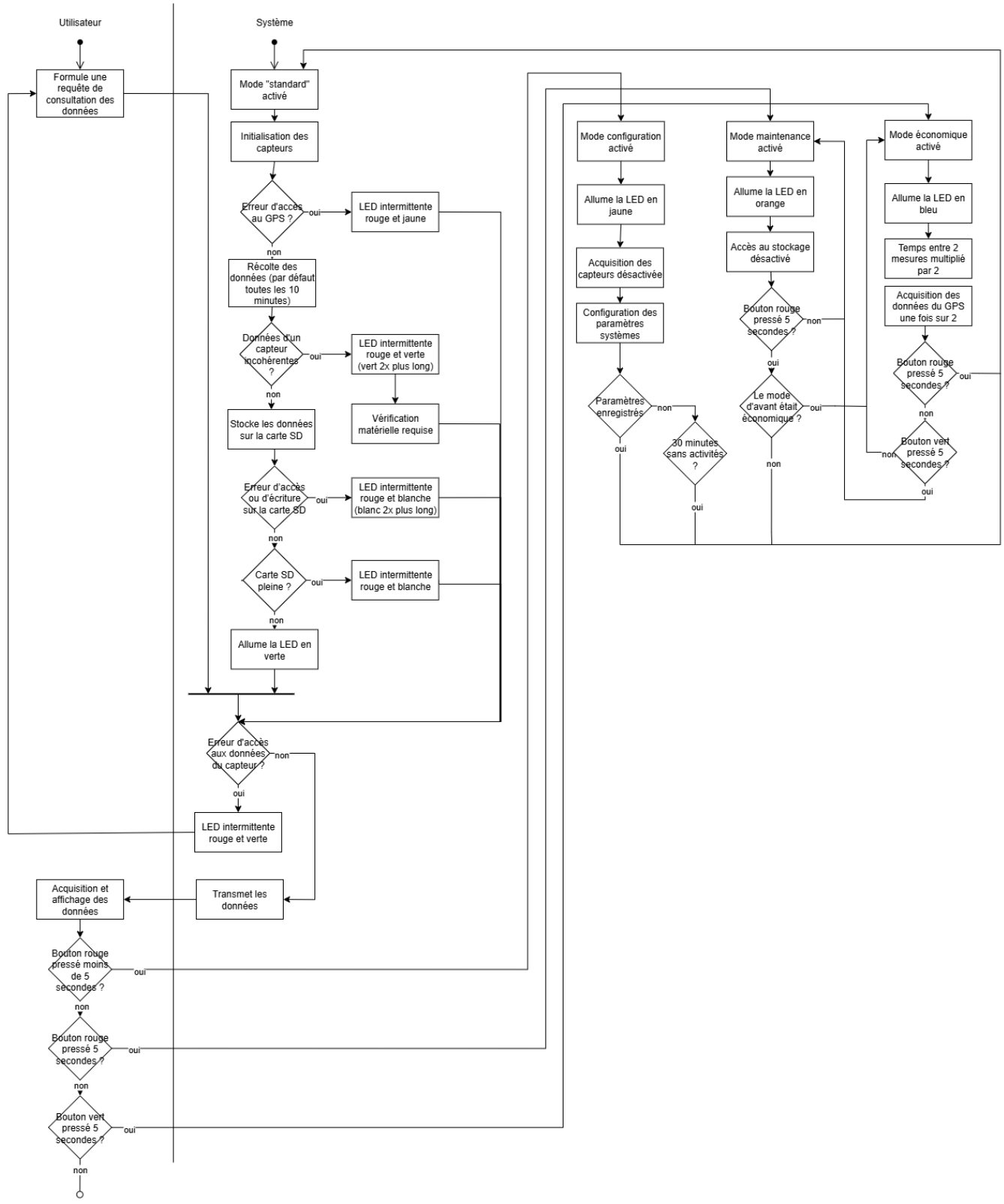
Acteurs (entités externes qui interagissent avec le système)

Relation

Include = le premier cas d'utilisation inclut le second et son issue dépend de la résolution du second (il est obligatoire)

Extend =cas d'utilisation A peut-être appelé au cours du cas d'utilisation de B (il est optionnel)

Diagramme d’activité :



Utilisateur : représente les interactions manuelles possibles

Système : contient toute la logique automatique de la station météo

Etape :

Phase d'initialisation et démarrage :

Au démarrage, le système active le mode standard puis procède à l'initialisation des capteurs. Cette phase critique vérifie le fonctionnement de tous les composants.
Le système vérifie avec signalisation visuelle via la LED RGB :

-Si le GPS n'est pas accessible, la LED clignote en rouge et jaune, signalant un problème

-Si l'un des capteurs renvoie des valeurs énormes, la LED clignote en rouge et vert, indiquant qu'une vérification matérielle est requise.

-Sans possibilité de sauvegarde, la LED clignote en rouge et blanc. C'est une erreur car les données ne peuvent être conservées.

-Lorsque l'espace de stockage est saturé, la LED clignote en rouge et blanc, nécessitant un remplacement ou de vider la carte.

Si toutes ces vérifications sont réussies, la LED s'allume en vert, confirmant que le système est prêt à fonctionner.

Ensuite nous avons la boucle principale :
Le système entre dans une boucle continue avec des vérifications régulières :
Le système surveille continuellement l'intégrité des données. En cas d'erreur d'accès aux données du capteur, la LED clignote en rouge et vert, alertant l'équipage d'un dysfonctionnement.

L'utilisateur peut déclencher l'affichage des données, qui sont ensuite transmises pour consultation ou pour une analyse externe.
Le système vérifie en permanence si les boutons poussoirs sont pressés.
Il y a trois types de pressions sont possibles :

- Bouton rouge pressé moins de 5 secondes : Action rapide ou validation
- Bouton rouge pressé 5 secondes ou plus : Action prolongée ou changement de mode
- Bouton vert pressé 5 secondes ou plus : Autre action ou retour au mode précédent

Pour la suite nous avons la gestion des modes opérationnels qui sont encadrés à droite, (encadré à droite).
Nous avons mis les caractéristiques techniques des modes de fonctionnement du système.

Notre diagramme d'activité montre la station météo embarqué conçue pour une utilisation autonome pour être installer dans un navire.

Légende :

Nœud initial ; c'est le début du processus

Décision

Action, c'est une étape de traitement

Flèche de transition

Absolument. Mes excuses, je ne peux pas générer directement de fichiers téléchargeables.

Cependant, j'ai rassemblé tout le contenu du livrable dans un seul bloc de texte. Vous pouvez simplement faire un **copier-coller** de tout ce qui se trouve ci-dessous dans un nouveau document Word. La mise en forme (titres, listes, blocs de code) sera conservée.

Livrable 2 : Architecture du Programme

Ce document présente l'architecture logicielle envisagée pour le microcontrôleur Arduino qui équipera la station météo embarquée. L'objectif est de définir une structure de code capable de gérer l'ensemble des fonctionnalités requises par le cahier des charges, notamment les différents modes de fonctionnement, l'acquisition des données des capteurs et la communication avec l'utilisateur.

Codes :

Générale:

Nous avons utilisé des commandes qui ne seront pas un blocage pour la suite comme
-Le code évite les delay() dans la boucle principale (loop()). Il utilise la fonction millis() pour gérer le temps (intervalle de mesure, timeouts, clignotement des LEDs, anti-rebond des boutons).
-Les boutons (Rouge et Vert) sont gérés par des interruptions (ISR). Celles-ci se contentent de lever un drapeau (volatile bool). La logique de traitement (anti-rebond, détection d'appui long) est ensuite gérée dans la boucle principale pour rester courte et rapide.

Variable :

- `Config`: Une structure contenant tous les **paramètres de configuration** du système.
- `LOG_INTERVAL` : Fréquence des mesures.
- `FILE_MAX_SIZE` : Taille maximale du fichier log (non utilisée dans ce code, mais prévue).
- `TIMEOUT_CAPTEUR` : Délai d'attente maximal pour une réponse de capteur (non utilisée dans ce code, mais prévue).
- `LUMIN_LOW` / `LUMIN_HIGH` : Seuils de luminosité (non utilisés dans ce code, mais prévus).
- `DataPoint`: Une structure représentant un **enregistrement de mesure unique**.
- `timestamp` : Date et heure de la mesure (format chaîne).
- `luminosite` : Valeur brute du capteur.
- `temperature_air` : Valeur de température.
- `gpsData` : Données GPS brutes (ou "NA").

Les 4 Modes

1. **MODE_STANDARD** : Mode par défaut. Lit tous les capteurs (GPS inclus) et enregistre sur la SD à l'intervalle `LOG_INTERVAL`.
2. **MODE_ECONOMIQUE** : L'intervalle de mesure est doublé (`LOG_INTERVAL * 2`) et le GPS n'est lu qu'une fois sur deux pour économiser l'énergie.

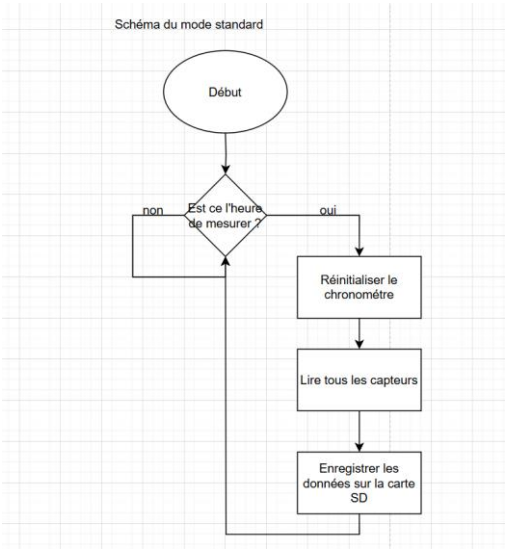
3. **MODE_CONFIGURATION** : Démarré en tenant le bouton Rouge. Permet de modifier la configuration via le port Série (ex: "RESET"). Quitte automatiquement après un timeout.
4. **MODE_MAINTENANCE** : Activé par un appui long. Affiche les données des capteurs sur le port Série (pour débogage) sans les enregistrer sur la SD.

Logigrammes des modes

Logigramme du Mode Standard

Ce schéma détaille le fonctionnement le plus courant : l'acquisition et l'enregistrement des données, en utilisant une temporisation non-bloquante.

Ce schéma décrit le cycle de mesure et d'enregistrement :



Logigramme du Mode Configuration :

Ce logigramme montre comment le système interagit avec l'utilisateur via la console série pour modifier les paramètres.

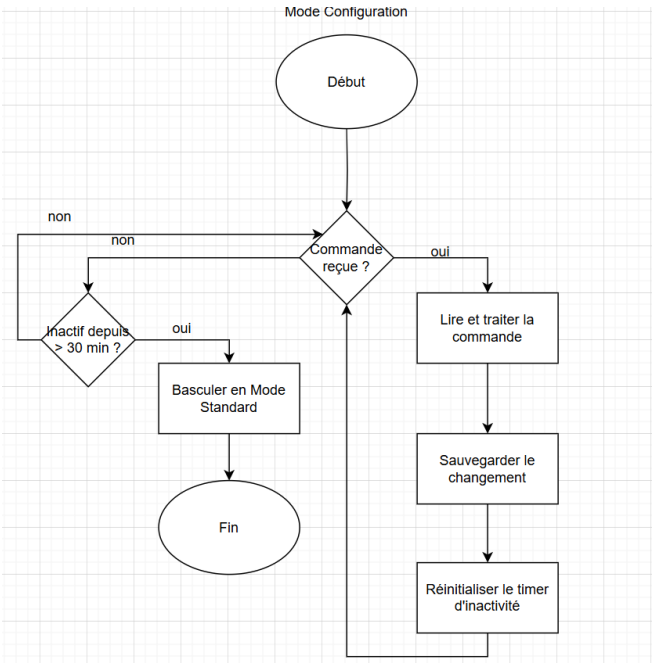


Schéma du Mode Maintenance :

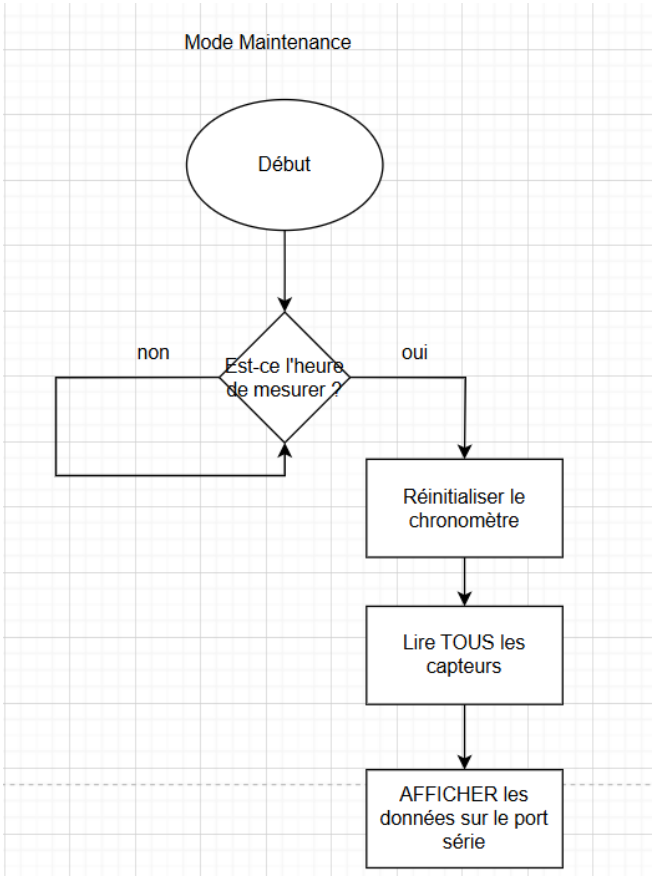
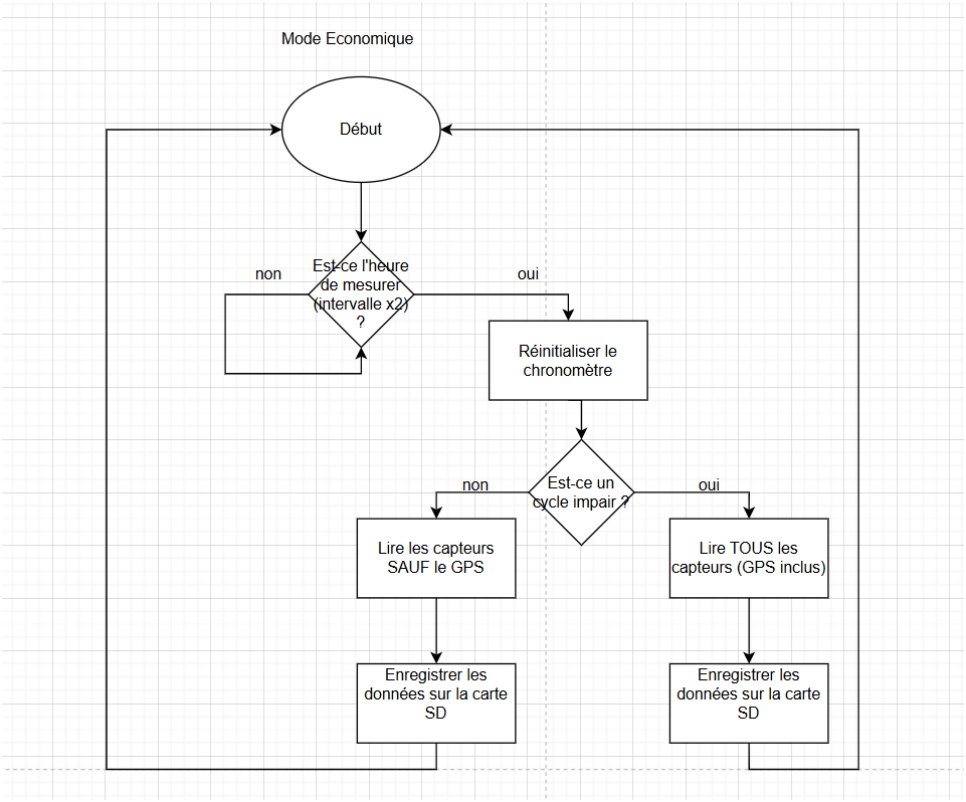


Schéma du Mode Économique :

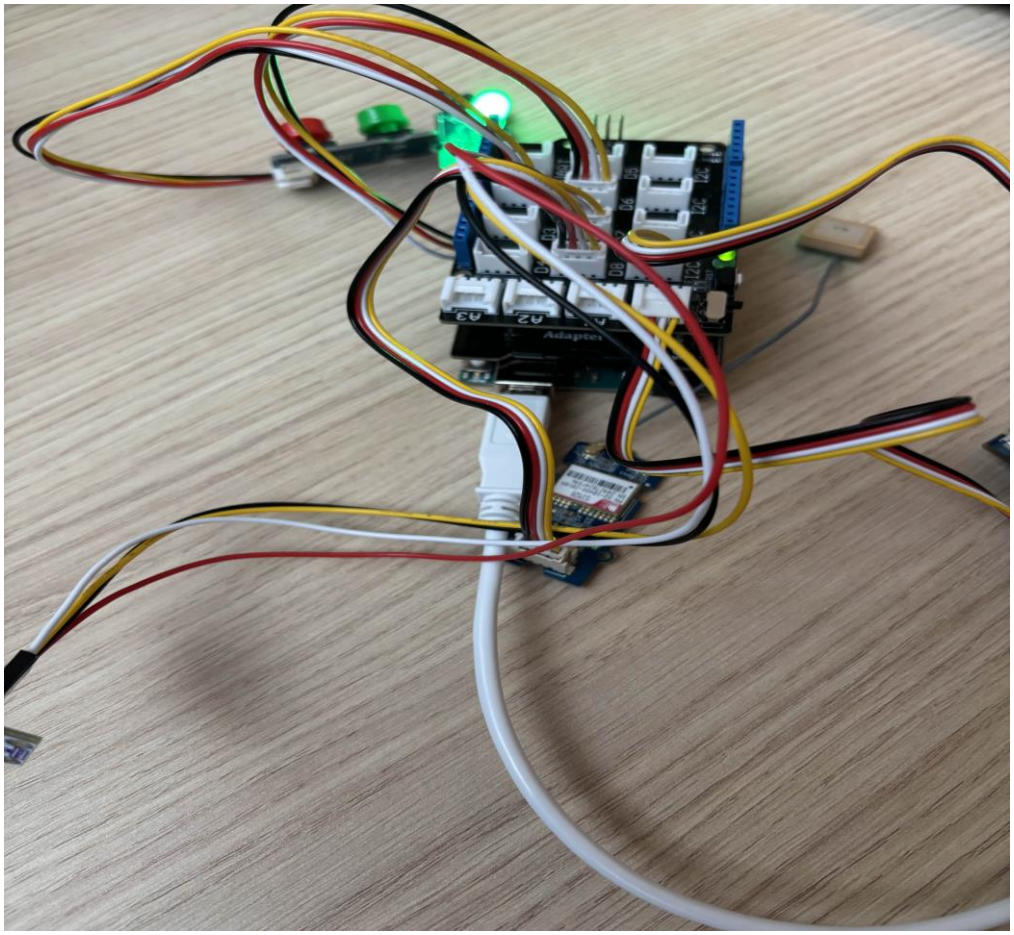
Ce schéma ajoute une décision supplémentaire pour déterminer s'il faut lire le GPS.

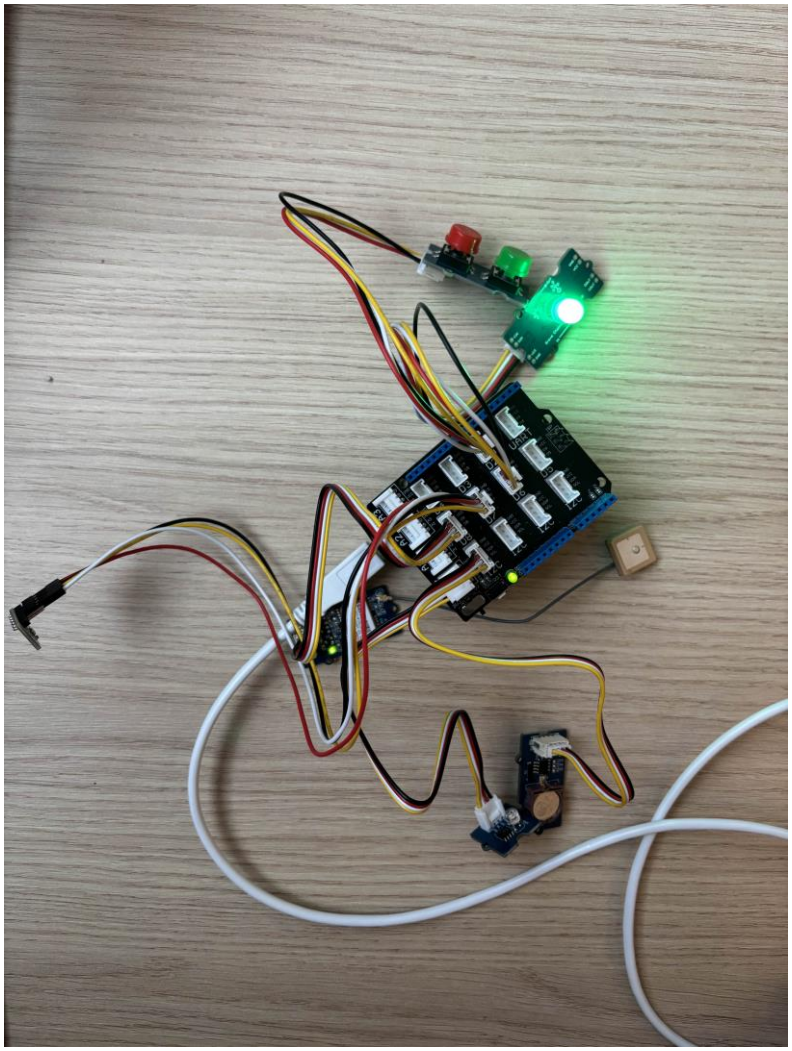


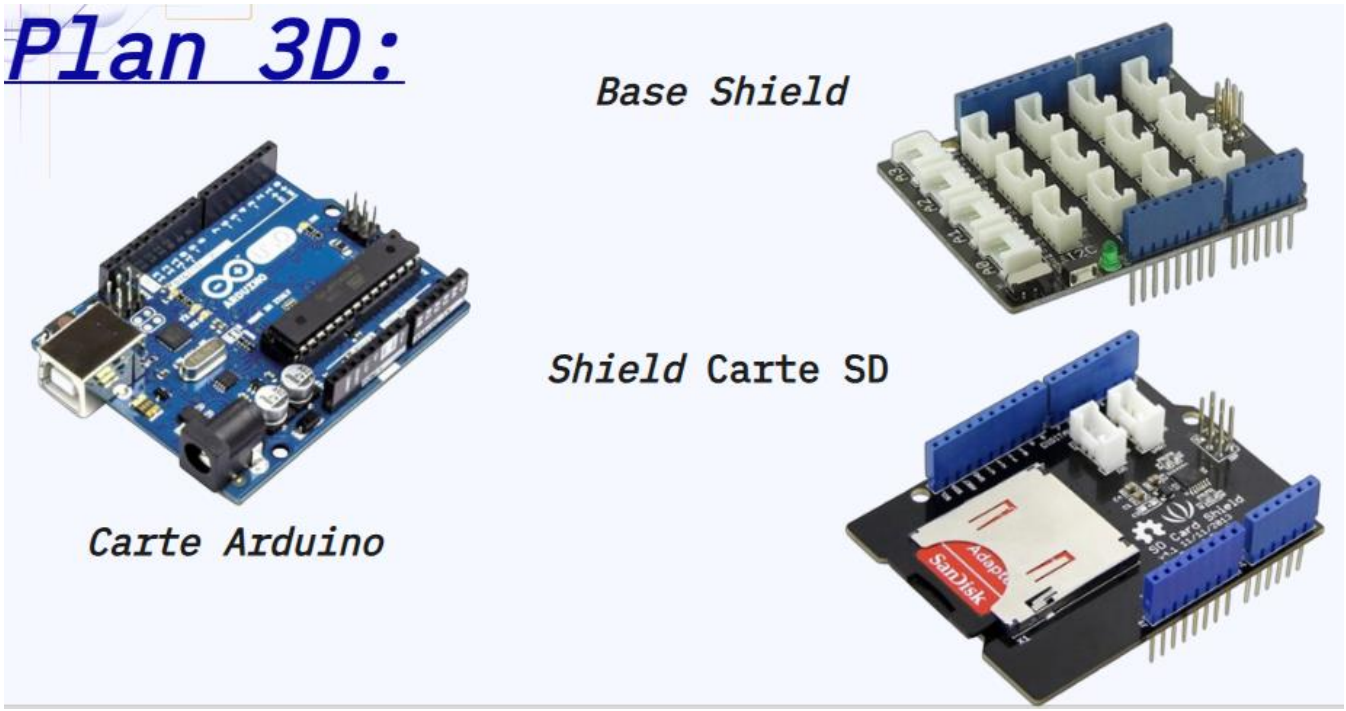
LIVRABLE 3

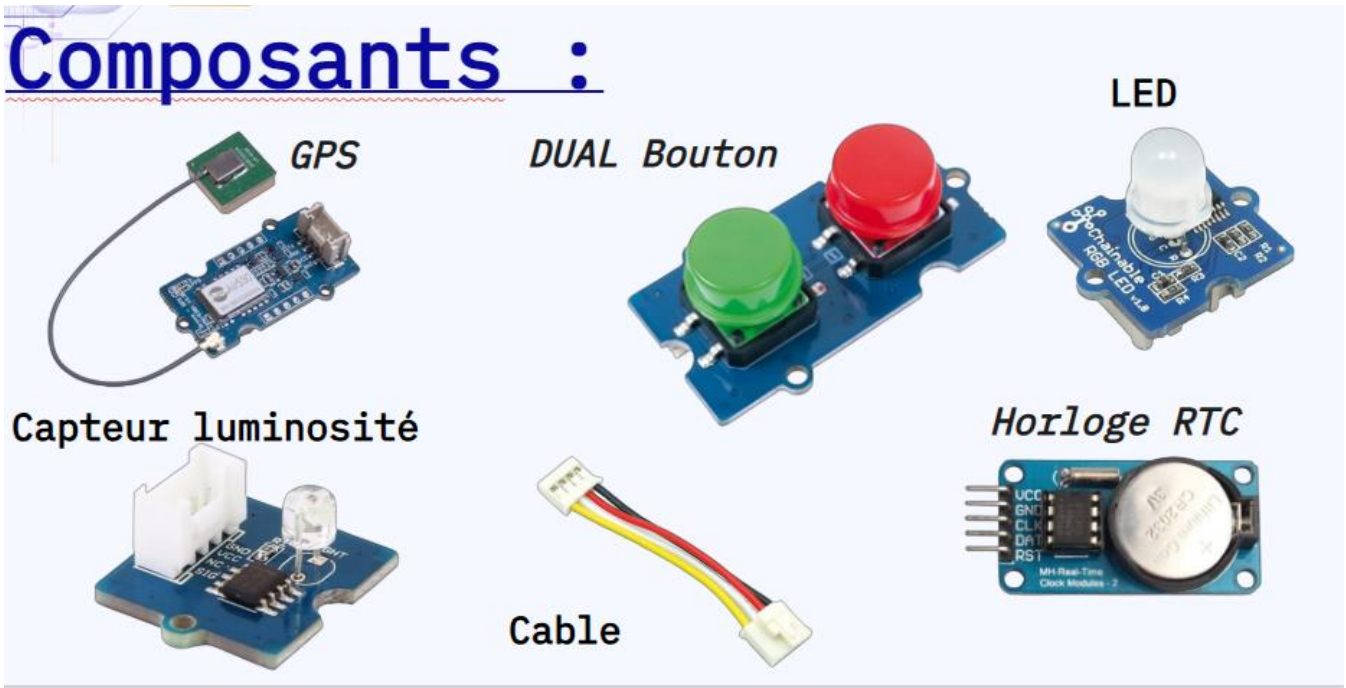
Programme complet de la carte :

Voici le plan de notre carte détaille avec tous nos capteurs intégrer :









Script de compilation/téléversement :

Pour pouvoir utiliser notre code, nous devons tout d'abord le compiler, puis le téléverser.
Cependant, avec une carte Arduino Uno, ces étapes se font automatiquement.

Dans notre cas, nous allons procéder à ces étapes manuellement, car nous ne disposons que d'un terminal, comme cela est demandé dans notre cahier des charges :

Nous installons tous d'abord :

sudo apt install arduino-cli

Puis nous compilons notre code :

arduino-cli compile --fqbn arduino:avr:uno mon_code

Mon_code est en .ino

Puis nous téléversons le code :

arduino-cli upload -p /(port de serie de la carte) --fqbn arduino:avr:uno mon_code

Cette ligne est pour afficher le port de la carte :

arduino-cli board list

- Le choix du compilateur était simple puisqu'Il configure automatiquement les options de compilation
- Il gère les **bibliothèques** Arduino .
- Il choisit le bon **core** selon la carte ,Uno par exemple .
- Il compile tous les fichiers nécessaires (.ino + .cpp + .h)

Livrable 4

Documentation Technique

Projet : Worldwide Weather Watcher (WWW)

Version : 1.0

Destinataires : Équipes de développement, ingénieurs de maintenance, futurs contributeurs au projet.

Fonctionnement Global du Système

Objectif

Le système WWW est une station météo embarquée prototype, conçue pour être installée sur des navires. Son objectif est de mesurer et d'enregistrer en continu des paramètres environnementaux (pression, température, hygrométrie, position GPS) afin de constituer une base de données pour la prévision de catastrophes naturelles.

Composants Principaux

Le système est articulé autour d'une **Carte Arduino** qui sert d'unité centrale de traitement. Elle gère :

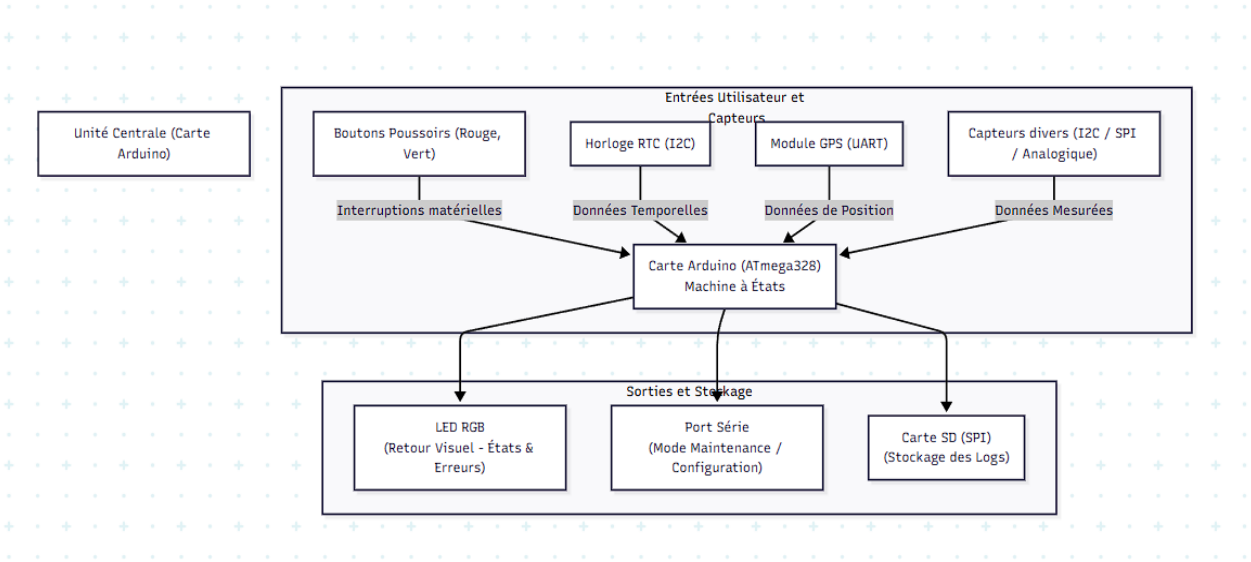
- **Les Entrées :**

- Horloge RTC (I2C) pour l'horodatage.
- Module GPS (UART) pour la géolocalisation.
- Capteurs divers (Pression, Température, Hygrométrie, Luminosité).
- Deux boutons poussoirs (Rouge, Vert) pour l'interaction utilisateur.

- **Les Sorties et le Stockage :**
  - Une LED RGB pour le retour visuel de l'état du système.
  - Un lecteur de carte SD (SPI) pour le stockage persistant des logs.
  - Une interface Série (UART) pour les modes Configuration et Maintenance.

Flux d'Information dans le sytéme (Schéma) :

Le schéma suivant illustre l'architecture des flux de données et de contrôle du système.



Unité Centrale

- **Carte Arduino (ATmega328)**
  → C'est le cœur du système.
  Elle exécute la logique du programme sous forme de machine à états, c'est-à-dire qu'elle change d'état en fonction des entrées (capteurs, boutons, etc.) et des conditions temporelles.

Entrées Utilisateur et Capteurs

Ces modules fournissent des informations à la carte Arduino :

1. **Boutons poussoirs (Rouge, Vert)**
   a. Servent à l'interaction manuelle de l'utilisateur (démarrer, arrêter, valider, etc.).
   b. Connectés via interruption matérielle pour une réaction rapide aux appuis.
2. **Horloge RTC (I2C)**
   a. Fournit des données temporelles précises (heure, date).
   b. Permet de dater les événements ou synchroniser les mesures.

3. **Module GPS (UART)**
    a. Fournit les données de position (coordonnées, vitesse, heure GPS, etc.).
4. **Capteurs divers (I2C / SPI / Analogique)**
    a. Mesurent différents paramètres physiques (température, pression, luminosité, etc.).
    b. Ces données mesurées sont utilisées par la machine à états pour la prise de décision.

Sorties et Stockage

La carte Arduino contrôle plusieurs modules de sortie et de stockage :

1. **LED RGB**
    a. Sert de retour visuel pour indiquer l'état du système ou signaler des erreurs.
2. **Port Série**
    a. Utilisé pour la configuration, la maintenance, ou le diagnostic.
    b. Permet une communication avec un PC ou un autre dispositif.
3. **Carte SD (SPI)**
    a. Sert au stockage des logs (enregistrement des mesures, événements, erreurs, etc.).

Flux d'Information

- Les entrées (capteurs, boutons, horloge, GPS) envoient des données vers la carte Arduino.
- L'Arduino traite ces données via sa machine à états et agit en conséquence :
    - Affiche un état via la LED RGB,
    - Transmet des informations via le port série,
    - Enregistre des données sur la carte SD.

En résumé

Ce schéma représente un système embarqué autonome centré sur une carte Arduino ATmega328, qui :

-Reçoit des données utilisateur et capteurs,

-Les traite selon un fonctionnement en machine à états,

-Et produit des sorties visuelles, de communication et de stockage.

Documentation Utilisateur

Guide de Démarrage Rapide : Station Météo WWW

Bienvenue ! Ce guide vous explique comment utiliser la station météo embarquée.

Qu'est-ce que cet appareil ?

C'est un "enregistreur" météo. Il sauvegarde automatiquement la température, la pression, la position GPS et d'autres données sur la carte SD.

Utilisation des Boutons :

Vous avez deux boutons à votre disposition qui va vous permettre de gère les modes configurations :

Mode	Méthode d'Accès	Description
Standard	Démarrage normal (sans bouton pressé).	Fait l'acquisition des données.
Configuration	Démarrage avec le bouton rouge pressé.	Permet de configurer les paramètres via l'interface série. L'acquisition est désactivée. Bascule en mode standard après 30 min d'inactivité.
Maintenance	Appui 5s sur bouton rouge (depuis Standard ou Éco).	Données des capteurs consultables sur port série. L'écriture sur carte SD est arrêtée. Permet de changer la carte SD en toute sécurité. Rebascule au mode précédent par appui 5s sur bouton rouge.
Économique	Appui 5s sur bouton vert (depuis Standard).	Économise la batterie. Acquisition GPS 1 mesure sur 2. Intervalle LOG_INTERVAL multiplié par 2. Rebascule en mode Standard par appui 5s sur bouton rouge.

Utilisation Erreur :

Vous avez plusieurs signaux lumineux qui corresponds à une erreur du système :

Signal Lumineux	État du Système
Modes Normaux (Continu)	
Vert Continu	Mode standard
Jaune Continu	Mode configuration
Bleu Continu	Mode économique
Orange Continu	Mode maintenance
Erreurs (Intermittent, 1Hz)	
(Rouge + Bleu)	Erreur d'accès à l'horloge RTC
(Rouge + Jaune)	Erreur d'accès aux données du GPS
Rouge + Verte (durées identiques)	Erreur accès aux données d'un capteur
Rouge + Verte (Vert 2x plus long)	Données reçues d'un capteur incohérentes
(Rouge + Blanc)	Carte SD pleine
(Rouge + Blanc) (Blanc 2x plus long)	Erreur d'accès ou d'écriture sur la carte SD

Ils ont été énumérés dans l'ordre pour vous compreniez mieux les problèmes.

Commandes du Mode Configuration

Liste des commandes textuelles disponibles sur l'interface série en mode configuration sans les commandes pour les capteurs.

Commande	Description
LOG_INTERVAL=10	Définit l'intervalle entre 2 mesures (10 minutes par défaut).
FILE_MAX_SIZE=4096	Définit la taille max (en octets) d'un fichier log avant archivage (4ko par défaut).
RESET	Réinitialisation de l'ensemble des paramètres à leurs valeurs par défaut.
VERSION	Affiche la version du programme et un numéro de lot.
TIMEOUT=30	Durée (en s) au bout de laquelle l'acquisition d'un capteur est abandonnée (30s par défaut).
CLOCK=...	Configuration de l'heure (HEURE:MINUTE:SECONDE).
DATE=...	Configuration de la date (MOIS,JOUR,ANNEE).
DAY=...	Configuration du jour de la semaine (MON, TUE, WED, etc.).
(Voir table 4)	Commandes spécifiques aux capteurs (ex: LUMIN=1, MIN_TEMP_AIR=-5).

Paramètres Configurables des Capteurs

Détail des paramètres modifiables en mode configuration pour les capteurs

Paramètre	Domaine	Défaut	Description	Exemple de Commande
LUMIN	{0,1}	1	Activation (1) / Désactivation (0) du capteur lumir	LUMIN=1
LUMIN_LOW	{0-1023}	255	Seuil de luminosité "faible".	LUMIN_LOW=200
LUMIN_HIGH	{0-1023}	768	Seuil de luminosité "forte".	LUMIN_HIGH=700
TEMP_AIR	{0,1}	1	Activation (1) / Désactivation (0) du capteur temp	TEMP_AIR=0
MIN_TEMP_AIR	{-40-85}	-10	Seuil T°C air (min) avant erreur capteur.	MIN_TEMP_AIR=-5
MAX_TEMP_AIR	{-40-85}	60	Seuil T°C air (max) avant erreur capteur.	MAX_TEMP_AIR=30
HYGR	{0,1}	1	Activation (1) / Désactivation (0) du capteur d'hyg	HYGR=1
HYGR_MINT	{-40-85}	0	T°C min en dessous de laquelle l'hygrométrie n'es	HYGR_MINT=0
HYGR_MAXT	{-40-85}	50	T°C max au-dessus de laquelle l'hygrométrie n'est	HYGR_MAXT=50
PRESSURE	{0,1}	1	Activation (1) / Désactivation (0) du capteur de pr	PRESSURE=0
PRESSURE_MIN	{300-1100}	850	Seuil Pression HPa (min) avant erreur capteur.	PRESSURE_MIN=450
PRESSURE_MAX	{300-1100}	1080	Seuil Pression HPa (max) avant erreur capteur.	PRESSURE_MAX=1030

Assistance en cas de problème ou de problème spécifique :

Chef de Projet : Paul Calais

Membre du projet : Lucas Baye

Enzo Verdin

Veuillez les contacter au 802546536

Conclusion :

Nous avons réalisé notre station météo grâce à une carte Arduino et de nombreux capteurs permettant de capte des donnes que nous pouvons lire par la suite. Notre guide technique pour que personne non connaissant dans l'informatique puisse réaliser des mesures grâce à notre système.