

Capteur environnemental : comment mesurer le développement de la végétation ?

BUT Mesures Physiques— 2e année (spécialisation TI)

Encadrants : Aurore Brut, Jean-Pascal Dezalay

Auteurs : Evan Gonzalez, Yannis Fabre, Taha Ghissassi, Félix Grues, Sandro Barrot

Date : Octobre 2025

Projet tutoré – Compte rendu de séance n°1

Objectif de la séance :

- Définir les besoins fonctionnels du capteur NDVI.
- Identifier les grandeurs physiques à mesurer.
- Sélectionner les capteurs nécessaires.
- Concevoir le design mécanique (boîtier, diffuseur, protection).
- Déterminer le mode de transmission et de stockage des données.
- Poser les bases de l'interface logicielle et du traitement des mesures.

Choix des capteurs :

L'indice NDVI (Normalized Difference Vegetation Index) repose sur la comparaison de la réflectance entre le rouge et le proche infrarouge (NIR).

Nous utiliserons comme capteur optique le capteur : **SEN-14351**

D'autres capteurs complémentaires permettront de mesurer les paramètres environnementaux:

Température / humidité / pression GY-BME280 ≈ 10 €

Mouvement / accélération MPU6050 ≈ 6 €

Télétransmission des données

Calcul des réflectances et du NDVI.

Stockage dans une base de données.

Traitement et structure des données

Chaque enregistrement contiendra :

intensité rouge et NIR pour le capteur incident,

intensité rouge et NIR pour le capteur feuillage,

$$R(\lambda) = I_r(\lambda) - I_n \frac{\lambda}{\{I_i(\lambda) - I_n(\lambda)\}}$$

Calcul du NDVI :

$$\text{NDVI} = \frac{R_{\text{NIR}} - R_{\text{Rouge}}}{R_{\text{NIR}} + R_{\text{Rouge}}}$$

Interface de visualisation

L'interface permettra d'explorer et d'interpréter facilement les mesures recueillies.

Fonctionnalités prévues :

- évolution du NDVI dans le temps,
- suivi de l'état des capteurs,

Exports : téléchargement CSV filtré,

Design du capteur et conception mécanique

Le capteur doit être étanche, résistant, et optiquement fiable.

Structure générale : Up / Down

Up : Diffuseur

Down : Isolation / Protection

Le diffuseur ne doit pas bloquer le proche infrarouge. Il doit diffuser la lumière uniformément et durablement dans le temps.

Matériaux possibles

- Plastique opalin / polycarbonate diffusant
- PTFE expansé (Teflon blanc, type Spectralon)
- Dôme diffusant en plastique blanc

Formes du diffuseur :

- Dôme hémisphérique
- Cylindre ou demi-sphère

Références possibles :

-SCANGRIP Diffuser Cap Small

-Neewer Dôme Diffuser Flash Z1

Points de vigilance avant le choix :

- Transmission dans la bande proche IR (vérification de la transmission spectrale).
- Diffusion uniforme.
- Résistance aux rayures et aux chocs.
- Stabilité et vieillissement du matériau.

Isolation et protection :

-Boîte électrique étanche (ABS ou polycarbonate).

-Pose d'un joint silicone autour de la fenêtre ou du dôme.

-Fenêtre optique transparente et antireflet sur la face supérieure (côté ciel), avec le diffuseur juste en dessous.

-Sur la face latérale, petite ouverture possible avec une vitre transparente.

-Peindre la boîte en blanc pour réduire l'échauffement au soleil.

Schéma du design du capteur:

Le schéma réalisé durant la séance illustre la structure verticale du dispositif :

Partie supérieure : Diffuseur (Up) → capte la lumière incidente et assure une diffusion homogène.

Partie intermédiaire : Zone d'isolation / protection (Down) → protège les capteurs et limite les interférences lumineuses.

Partie inférieure : Boîtier électronique → contient la carte Arduino, les capteurs et les connexions.

L'ensemble vise à :

- garantir la stabilité mécanique,
- éviter les fuites lumineuses,
- assurer une bonne transmission NIR.

(Le schéma du design sera inséré ici dans la version finale du rapport.)

Principe physique et sélection des capteurs optiques Végétation saine

Une végétation saine absorbe fortement dans le rouge et réfléchit beaucoup dans le proche infrarouge (NIR). Cette différence spectrale permet de suivre la santé et le développement de la végétation via le NDVI.

Sélection de capteurs optiques

Conclusion

Cette première séance a permis d'établir un cahier des charges complet et de structurer le projet. Les capteurs nécessaires ont été sélectionnés, les principes de télétransmission et de traitement des données définis, et le design mécanique (diffuseur, boîtier, protection) a été étudié.

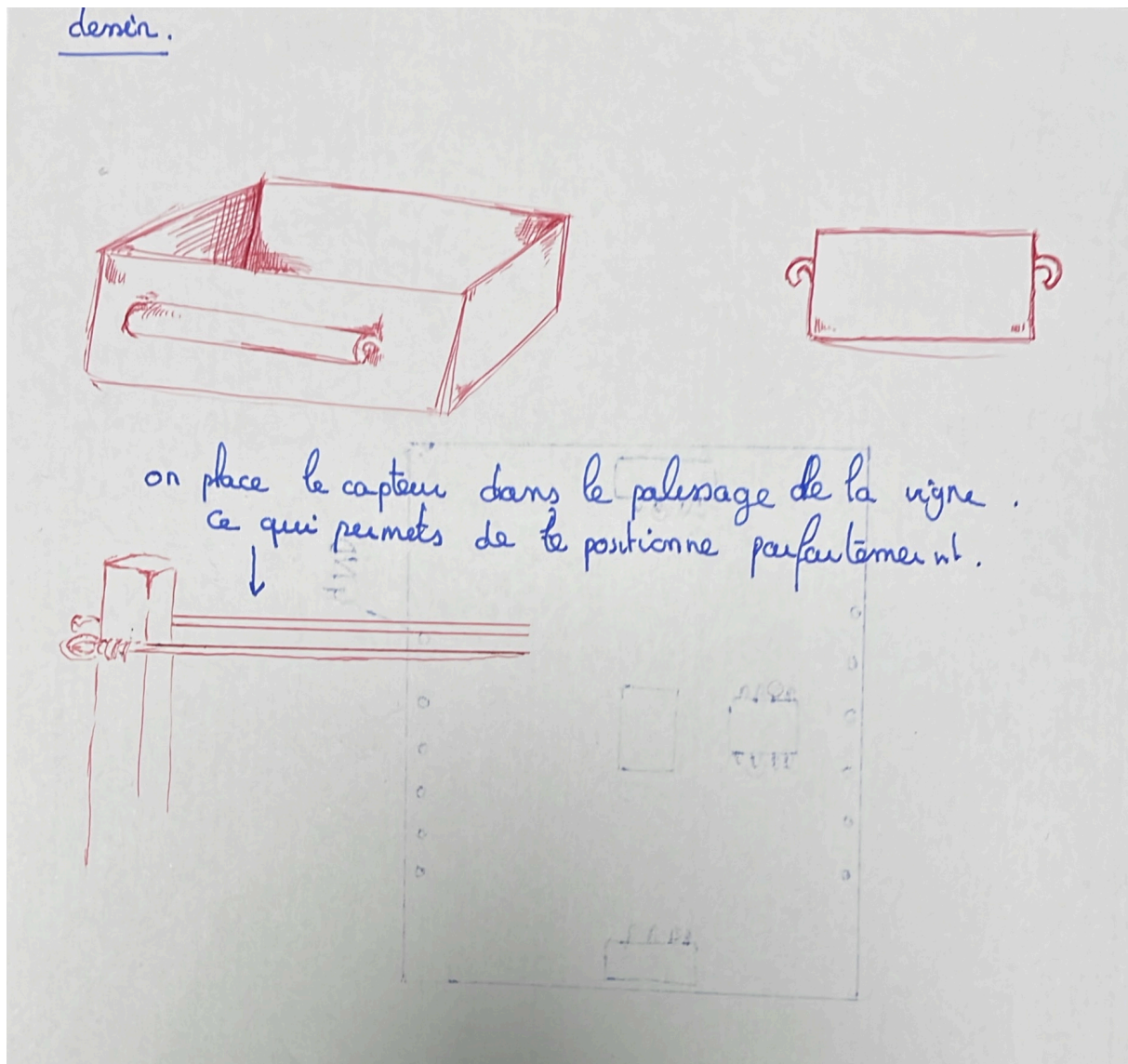


Schéma capteur final

Compte rendu – Deuxième séance du projet tutoré

Objectifs de la séance

- Réaliser le design complet de la carte électronique à partir des composants sélectionnés lors de la première séance.
- Répertorier précisément les caractéristiques électriques et les contraintes de chaque composant afin d'éviter toute erreur lors de la mise sous tension (tensions maximales, intensités, polarités, brochages, etc.).
- Approfondir la recherche et la prise en main du logiciel **Eagle Autodesk** pour la conception du schéma électronique et le routage de la carte.

Description de la séance

Cette deuxième séance a été consacrée à la conception matérielle du système électronique. Après avoir défini les besoins fonctionnels et sélectionné les capteurs lors de la première séance, l'équipe s'est concentrée sur la partie **électronique du projet**.

L'objectif principal était de collecter les fiches techniques de chaque composant (capteurs, microcontrôleur, convertisseurs, connecteurs, etc.) et d'en extraire les caractéristiques clés :

- Tensions d'alimentation minimales et maximales,
- Courants consommés,
- Niveaux logiques des signaux,
- Protocoles de communication,
- Brochages précis et contraintes de câblage.

Cette phase est essentielle pour éviter tout risque de détérioration lors des premiers essais sous tension. Les données recueillies serviront à établir un schéma clair et fiable sur le logiciel **Eagle Autodesk**, qui sera ensuite utilisé pour le routage du futur circuit imprimé (PCB).

Une attention particulière a été portée à la compatibilité électrique entre les capteurs et la carte Arduino. Tous les capteurs utilisés communiquent via le bus I²C, ce qui simplifie le câblage (liaison à deux fils : SDA pour les données et SCL pour l'horloge). Cependant, il est nécessaire de vérifier que les niveaux logiques sont compatibles :

- L'Arduino UNO utilise du 5 V logique,
- Certains capteurs, comme le **AS7263** et le **BME280**, fonctionnent uniquement en 3,3 V logique.

Pour cette raison, un translateur de niveau peut être nécessaire afin d'éviter une surtension sur les entrées des capteurs sensibles.

Étude et câblage des capteurs

AS7263 — Capteur de spectroscopie proche infrarouge

Le **AS7263** est un capteur multispectral mesurant la réflectance dans six bandes du proche infrarouge (NIR), entre 610 nm et 860 nm. Dans le cadre du projet NDVI, il est utilisé pour mesurer la réflectance dans le proche infrarouge (≈ 810 nm), nécessaire au calcul de l'indice de végétation.

Caractéristiques principales :

- Tension d'alimentation : 2,7 V à 3,6 V (idéalement 3,3 V).
- Consommation : environ 5 mA en fonctionnement.
- Interface : I²C (SDA, SCL) ou UART.
- Adresse I²C : 0x49 (modifiable sur certaines versions).

Branchement avec Qwiic : Le module breakout SparkFun intègre un connecteur Qwiic, ce qui permet de relier directement les lignes d'alimentation et de communication (VCC, GND, SDA, SCL) sans soudure.

Branchement sans Qwiic : Si le câblage est réalisé manuellement, les connexions sont :

- VCC → 3,3 V
- GND → GND
- SDA → A4
- SCL → A5

Si la carte Arduino tire les lignes SDA/SCL à 5 V, il faut utiliser un translateur de niveau logique ou vérifier que les résistances de **pull-up** sont reliées au 3,3 V.

Bibliothèque utilisée : SparkFun_AS726X

MPU6050 — Capteur d'accéléromètre et gyroscope

Le **MPU6050** permet de mesurer les accélérations linéaires et la vitesse angulaire sur trois axes. Dans le projet, il est prévu pour mesurer d'éventuelles inclinaisons ou vibrations du support afin d'évaluer la stabilité du capteur NDVI.

Caractéristiques principales :

- Tension d'alimentation : 3,3 V à 5 V.
- Interface : I²C (adresse 0x68 par défaut, 0x69 si ADO = 1).
- Consommation : environ 3,9 mA.
- Sensibilité : ± 2 g à ± 16 g (accéléromètre), ± 250 à ± 2000 °/s (gyroscope).

Branchement :

- VCC → 3,3 V
- GND → GND
- SDA → A4
- SCL → A5
- ADO → GND (adresse I²C = 0x68)

Des résistances de pull-up à 3,3 V sont recommandées pour stabiliser la communication I²C.

BME280 — Capteur de température, pression et humidité

Le **BME280** est un capteur environnemental mesurant la température, la pression atmosphérique et l'humidité relative. Il complète le capteur NDVI en fournissant des données de contexte permettant d'interpréter la réflectance de la végétation (influence de l'humidité de l'air, par exemple).

Caractéristiques principales :

- Tension d'alimentation : 3,3 V (tolère jusqu'à 5 V sur certaines versions breakout).
- Interface : I²C (adresse 0x76 ou 0x77).
- Consommation : environ 0,6 mA.

Branchement :

- VCC → 3,3 V
- GND → GND
- SDA → A4
- SCL → A5

Ce capteur est directement compatible avec le bus I²C commun aux autres capteurs.

Vérification de compatibilité électrique

Tous les capteurs utilisent le même protocole I²C, ce qui permet un bus partagé sur les broches A4 (SDA) et A5 (SCL). Chaque capteur possède une adresse I²C unique, ce qui évite les conflits lors de la communication simultanée. Les tensions d'alimentation ont été uniformisées à 3,3 V pour assurer la sécurité et la compatibilité des composants les plus sensibles (AS7263 et BME280).

Un régulateur de tension pourra être ajouté sur la carte si la carte Arduino n'intègre pas de sortie 3,3 V stable suffisante.

Conclusion

Cette deuxième séance a permis de :

- Finaliser la sélection et l'étude des composants électroniques,
- Vérifier la compatibilité électrique et logique entre les capteurs et la carte microcontrôleur,
- Commencer la modélisation du schéma électronique sur **Eagle Autodesk**,
- Préparer le futur routage du PCB.

Compte rendu – Troisième séance du projet tutoré

Choix et composante de la carte Arduino

Carte Arduino Uno ou Arduino nano 33 Arduino Uno: Carte moins chère plus répandue, mais domaine de tension à 5V donc risque de surtension sur les capteurs en 3V (AS7263 par exemple). On va donc plutôt sélectionner la Nano 33 car elle est plus adaptée à nos capteurs bien qu'elle coûte un peu plus cher.

Choix du clock: DS3231. Sa précision est très élevée (± 2 ppm $\approx \pm 1$ min/an), communication I²C, Alimentation 3,3 V / 5 V. Elle possède un capteur de température intégré pour compensation thermique

Choix de la carte SD: microSDHC de 8 Go ou 16 Go, formatée en FAT32. Elle n'est pas coûteuse, mais suffisante et fiable.

Arduino:

https://fr.rs-online.com/web/p/arduino/1927585?cm_mmc=FR-PLA-DS3A-_-bing-_-PLA_FR_FR_Catch+All-_-Fallback+-+Taxonomy+Change-_-1927585&matchtype=e&pla-4575067904093894&cq_src=google_ads&cq_cmp=554644914&cq_term=&cq_plac=&cq_net=o&cq_plt=gp&msclkid=ca198b785c541e327a70e3827868b35c&gclid=ca198b785c541e327a70e3827868b35c&gclsrc=3p.ds&gad_source=7&gad_campaignid=20595363253

Clock : https://fr.aliexpress.com/item/1005009475567776.html?src=bing&aff_short_key=UneMJZVf&aff_platform=true&isdl=y&albch=shopping&acnt=135095331&isdl=y&albcpr=554358948&albag=1309518793593597&slnk=&trgt=pla-4585444531847149&plac=&crea=81845014910635&netw=o&device=c&mtctp=e&utm_source=Bing&utm_medium=shopping&utm_campaign=PA_Bing_FR_PLA_PC_ALL_ECPC_20230403&utm_content=Ad%20group%20%231&utm_term=horloge%20DS3231&msclkid=4fc41bc7782713f9d346733d4ab154d6&gatewayAdapt=glo2fra

Sd:

- Recherche sur Eagle
- Réflexion sur la suite du projet
- Mise en forme du compte rendu et apprentissage du langage de codage Typst

Compte rendu – Quatrième séance du projet tutoré

Schéma du capteur AS7263

Applications

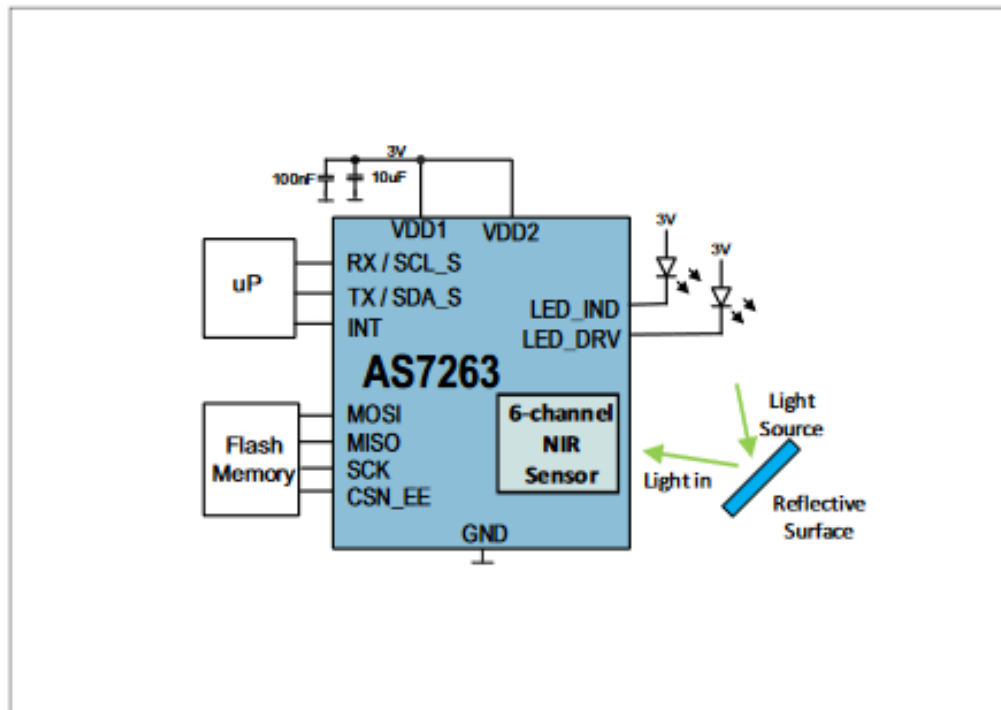
The AS7263 applications include:

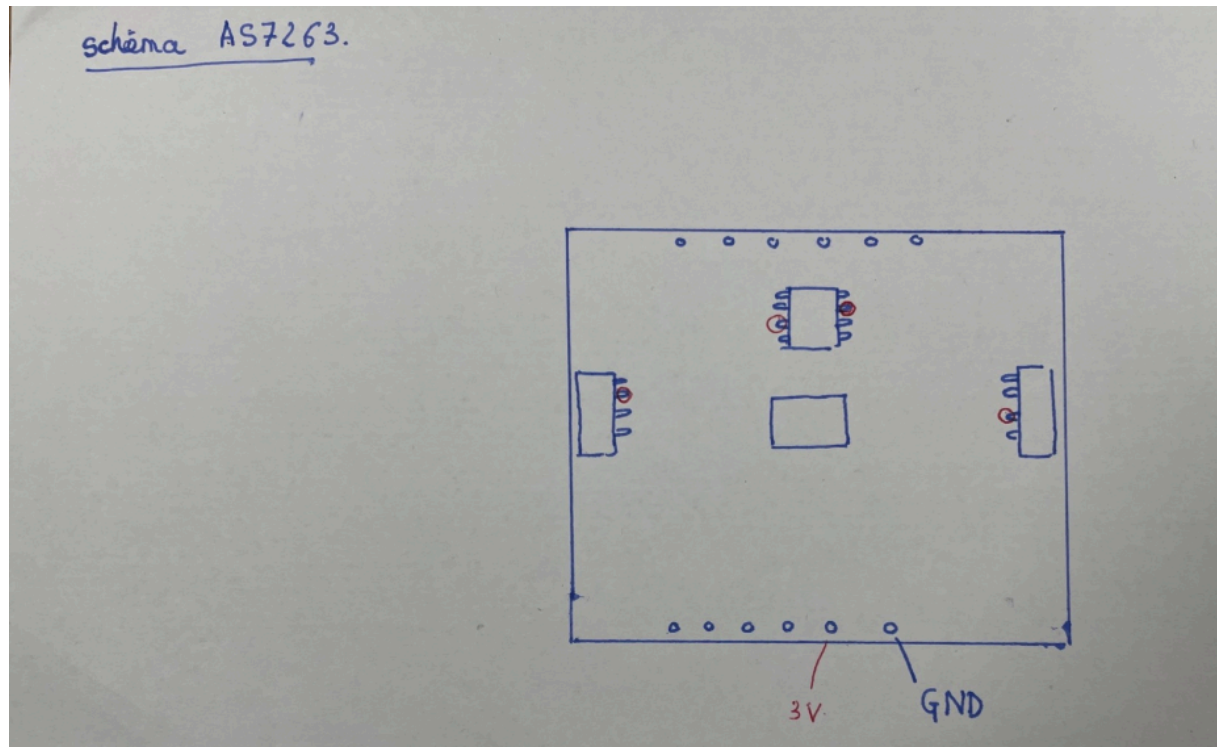
- Product authentication
- Bank note/document validation
- Chemical analysis
- Food/beverage safety

Block Diagram

The functional blocks of this device are shown below:

Figure 2:
AS7263 NIR Spectral_ID System





Programme Arduino : installer bibliothèques:

- Wire.h
- SPi.h
- SD.h
- RTCLib
- Adafruit_BME280
- SparkFun_AS7265X
- EEPROM

Compte rendu – Cinquième séance du projet tutoré

Programme :

sketch_nov13b

```
#include <Wire.h>
#include "AS726X.h"    // Bibliothèque SparkFun AS726X

AS726X sensor;        // Objet capteur (AS7262 ou AS7263)

void setup() {
    Serial.begin(115200);
    Wire.begin();
    delay(500);

    Serial.println();
    Serial.println(F("=== Initialisation du capteur AS726X ==="));

    // Initialisation du capteur
    if (!sensor.begin()) {
        Serial.println(F("ERREUR : aucun capteur AS7262/AS7263 detecte !"));
        Serial.println(F("-> Verifie : VIN=3.3V, GND=GND, SDA=A4, SCL=A5"));
        while (1) delay(1000); // Bloque ici si capteur non détecté
    }

    // Identification du type de capteur
    byte version = sensor.getVersion();
    if (version == SENSORTYPE_AS7262) {
        Serial.println(F("Capteur detecte : AS7262 (Visible)."));
    }
    else if (version == SENSORTYPE_AS7263) {
        Serial.println(F("Capteur detecte : AS7263 (Near IR)."));
    }
    else {
        Serial.print(F("Type inconnu (code=)"));
        Serial.print(version);
        Serial.println(F(")."));
    }

    // Réglages
    sensor.setGain(2);          // Gain : 0=1x,1=3.7x,2=16x,3=64x
    sensor.setIntegrationTime(50); // Temps d'intégration (~140 ms)
    sensor.disableIndicator();   // Couper la LED interne

    Serial.println(F("Configuration OK. Lecture des mesures..."));
    Serial.println();
}

void loop() {
    sensor.takeMeasurements(); // Mesure complète

    // Identifie le type du capteur
```

```

byte version = sensor.getVersion();

if (version == SENSORTYPE_AS7263) {
  // --- AS7263 - Proche infrarouge ---
  float R = sensor.getCalibratedR(); // 610 nm
  float S = sensor.getCalibratedS(); // 680 nm (Rouge)
  float T = sensor.getCalibratedT(); // 730 nm
  float U = sensor.getCalibratedU(); // 760 nm
  float V = sensor.getCalibratedV(); // 810 nm (NIR)
  float W = sensor.getCalibratedW(); // 860 nm (NIR)

  Serial.print(F("AS7263 | "));
  Serial.print("R(610)="); Serial.print(R, 3);
  Serial.print(" S(680)="); Serial.print(S, 3);
  Serial.print(" T(730)="); Serial.print(T, 3);
  Serial.print(" U(760)="); Serial.print(U, 3);
  Serial.print(" V(810)="); Serial.print(V, 3);
  Serial.print(" W(860)="); Serial.print(W, 3);

  // NDVI = (NIR - RED) / (NIR + RED)
  float ndvi = 0.0;
  if ((S + V) != 0) {
    ndvi = (V - S) / (V + S);
  }
  Serial.print(" | NDVI="); Serial.print(ndvi, 3);
}
else if (version == SENSORTYPE_AS7262) {
  // --- AS7262 (visible) si jamais vous testez l'autre version ---
  float vio = sensor.getCalibratedViolet();
  float blu = sensor.getCalibratedBlue();
  float grn = sensor.getCalibratedGreen();
  float yel = sensor.getCalibratedYellow();
  float org = sensor.getCalibratedOrange();
  float red = sensor.getCalibratedRed();

  Serial.print(F("AS7262 | V=")); Serial.print(vio, 2);
  Serial.print(" B="); Serial.print(blu, 2);
  Serial.print(" G="); Serial.print(grn, 2);
  Serial.print(" Y="); Serial.print(yel, 2);
  Serial.print(" O="); Serial.print(org, 2);
  Serial.print(" R="); Serial.print(red, 2);
}

// Température interne
float tempC = sensor.getTemperature();

```

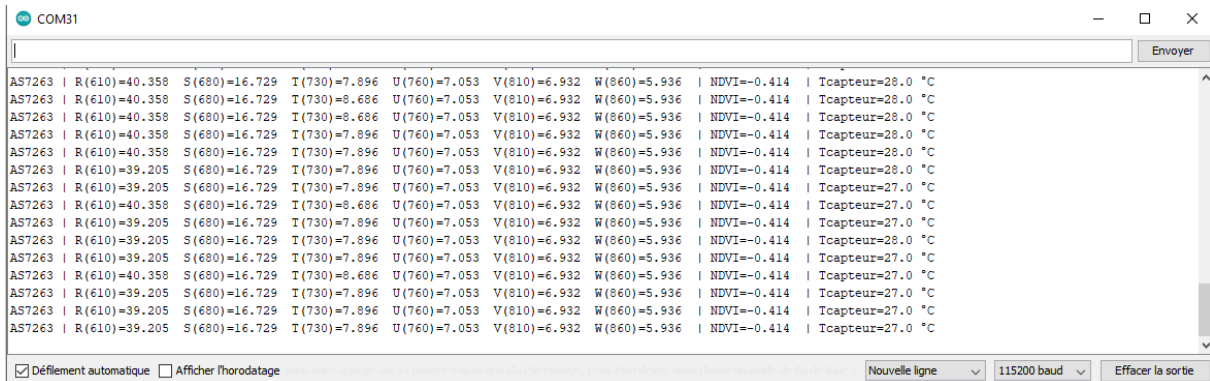
```

// Température interne
float tempC = sensor.getTemperature();
Serial.print(" | Tcapteur=");
Serial.print(tempC, 1);
Serial.print(" °C");

Serial.println();
delay(1000);
}

```

Rendu dans le moniteur série :



- R à W = intensité lumineuse détectée par le capteur dans chaque bande (610 → 860 nm).
- S(680) = canal rouge utilisé pour le NDVI.
- V(810) = canal NIR utilisé pour le NDVI.
- NDVI => $(V-S)/(V+S)$ = indice de végétation (ici négatif → pas de plante).
- Tcapteur = température interne du capteur.

Compte rendu – Sixième séance du projet tutoré

Objectifs de la séance

Cette sixième séance avait pour objectifs principaux :

- la finalisation de la conception mécanique du boîtier du capteur environnemental,
- la validation du fonctionnement des différents capteurs par des tests expérimentaux,
- l'avancement de la conception du schéma électronique sous Eagle.

1. Finalisation du boîtier sous Fusion 360

Lors de cette séance, le travail de conception du boîtier a été finalisé à l'aide du logiciel Fusion 360. Un paramétrage du modèle a été mis en place afin de rendre la conception plus flexible et facilement modifiable. Les dimensions principales du boîtier (épaisseur des parois, emplacements des capteurs, passages des câbles et points de fixation) ont été définies à l'aide de paramètres, permettant une adaptation rapide en cas d'évolution du projet.

Cette étape a permis de vérifier l'intégration mécanique de l'ensemble des composants (capteurs, carte Arduino, modules annexes) et d'anticiper les contraintes liées à l'assemblage ainsi qu'à l'utilisation sur le terrain.

2. Tests des différents capteurs

Une phase de tests des capteurs a été réalisée afin de vérifier leur bon fonctionnement avant leur intégration définitive dans le système. Chaque capteur a été alimenté et testé individuellement à l'aide de l'Arduino afin de s'assurer :

de la bonne communication, notamment sur le bus I²C,

de la cohérence des données mesurées,

de l'absence de conflits d'adresses ou de dysfonctionnements électriques.

Ces essais ont permis de valider la chaîne de mesure et de confirmer que les capteurs répondent aux attentes du projet.

3. Avancement du schéma électronique sous Eagle

La séance a également été consacrée à l'avancement du schéma électronique sous Eagle. Le travail réalisé ne relevait pas d'une simple prise en main du logiciel, mais d'une phase de conception déjà bien engagée. Le schéma a été structuré autour des différents blocs fonctionnels du système (alimentation, microcontrôleur, capteurs, communication), en veillant à la cohérence des connexions et au respect des contraintes électriques.

Cette étape constitue une base solide pour la suite du projet, notamment pour le routage du circuit imprimé et la réalisation du PCB.

Conclusion

Cette sixième séance a permis d'avancer de manière significative sur plusieurs aspects du projet. La finalisation du boîtier, la validation expérimentale des capteurs et l'avancement du schéma électronique sous Eagle marquent une étape importante vers l'intégration complète du système et la fabrication du circuit imprimé.

Compte rendu – Septième séance du projet tutoré

Objectifs de la séance

Cette septième séance avait pour objectifs :

- la finalisation du schéma électronique sous Eagle,
- le début du routage du circuit imprimé (PCB),
- la poursuite du développement du code Python,
- l'avancement du montage électronique du système.

1. Finalisation du schéma électronique

Le schéma électronique a été finalisé sous Eagle au cours de cette séance. L'ensemble des composants nécessaires au projet (microcontrôleur, capteurs, modules de communication et d'alimentation) a été intégré et organisé de manière structurée, en respectant les contraintes électriques et fonctionnelles du système.

Cette étape a permis de valider l'architecture globale de la carte et de préparer le passage vers la phase de routage du circuit imprimé.

2. Début du routage du PCB

Une fois le schéma validé, le travail s'est poursuivi par le début du routage du PCB. Le placement des composants a été réfléchi afin d'optimiser :

- la lisibilité du routage,
- la longueur des pistes,
- la gestion des alimentations et des signaux sensibles.

Les premières pistes ont été routées en tenant compte des règles de conception (largeur de pistes, espacements, organisation des masses), constituant une première version du circuit imprimé.

3. Avancement du code Python

En parallèle, les autres membres du groupe ont poursuivi le développement du code Python. Ce travail porte principalement sur la récupération, le traitement et l'exploitation des données issues des capteurs, ainsi que sur la structuration du programme pour permettre une acquisition fiable et exploitable.

Ces avancées logicielles sont essentielles pour assurer la cohérence entre l'acquisition matérielle et le traitement des données.

4. Poursuite du montage électronique

Le montage électronique du système a également été poursuivi. Les différents composants ont été assemblés et testés afin de valider le fonctionnement global du montage et d'identifier d'éventuels problèmes avant l'intégration finale sur le PCB.

Conclusion

Cette septième séance a permis des avancées majeures sur le projet. La finalisation du schéma électronique et le début du routage du PCB constituent une étape clé vers la fabrication de la carte. En parallèle, les progrès réalisés sur le code Python et le montage électronique renforcent la cohérence globale du système et préparent les prochaines phases d'intégration et de tests.

Compte rendu – Travail en autonomie (période de vacances)

Contexte

Suite à la septième séance du projet tutoré, une période de vacances a eu lieu. Durant cette période, le travail s'est poursuivi en autonomie afin de maintenir l'avancement du projet et de respecter le planning initial.

1. Finalisation du PCB

Le travail principal réalisé en autonomie a porté sur la finalisation du circuit imprimé (PCB) sous Eagle. Le routage a été complété en tenant compte des contraintes de conception définies précédemment, notamment :

- l'organisation des alimentations et des masses,
- la séparation des signaux sensibles,
- l'optimisation du passage des pistes et la réduction de leur longueur.

Les dernières vérifications ont été effectuées afin de s'assurer de la cohérence entre le schéma électronique et le routage, ainsi que du respect des règles de conception avant fabrication.

2. Vérifications et préparation à la fabrication

Une attention particulière a été portée aux points suivants :

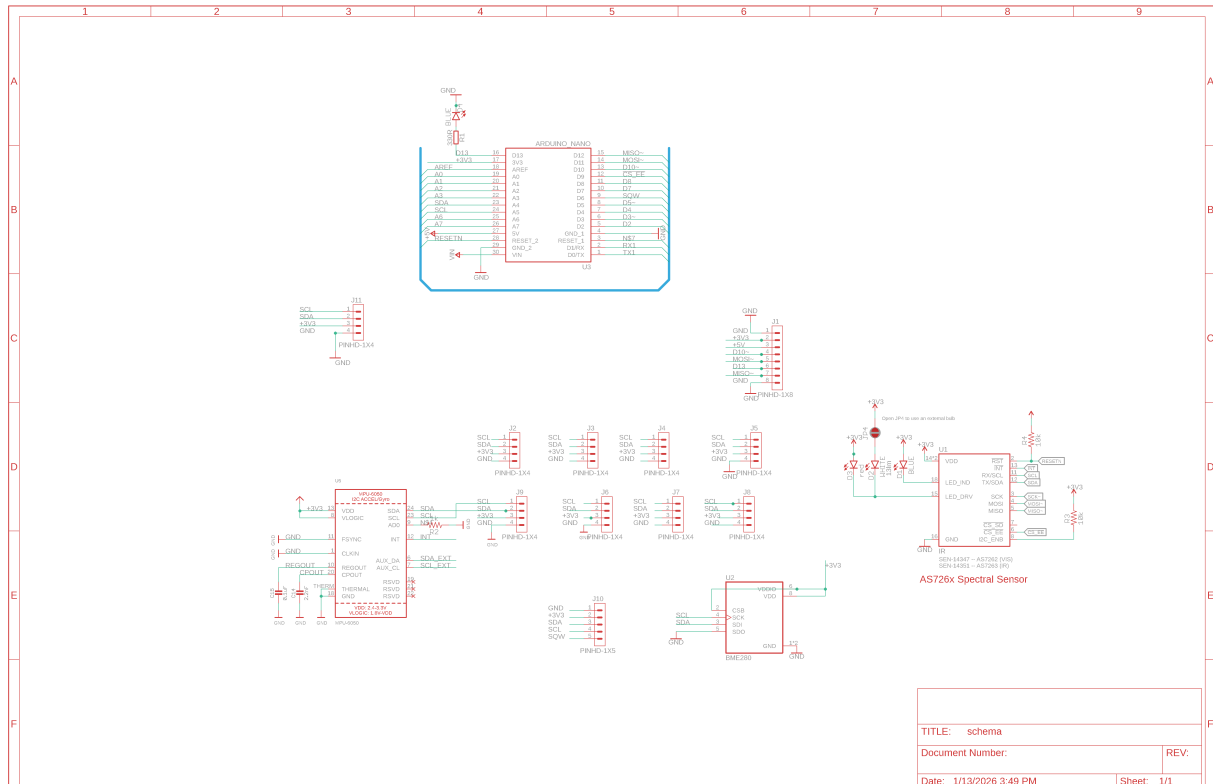
vérification des connexions et des empreintes des composants,
contrôle des erreurs électriques (ERC) et de routage (DRC),

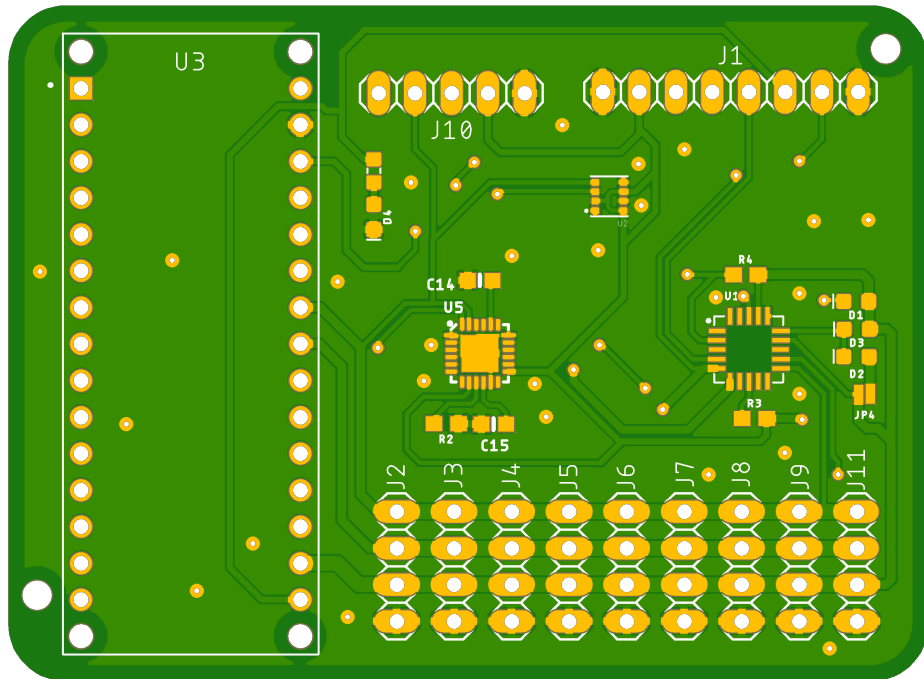
préparation des fichiers nécessaires à la fabrication du PCB.

Cette phase a permis d'aboutir à une version finalisée et exploitable de la carte électronique.

Conclusion

La période de travail en autonomie a permis de finaliser le PCB, étape essentielle du projet. Ce travail assure une continuité avec les séances encadrées précédentes et prépare la suite du projet, notamment la fabrication de la carte, l'assemblage et les tests expérimentaux.





Compte rendu – Séance d’organisation et de répartition finale du projet

Contexte

À l’issue de la finalisation du PCB réalisée en autonomie durant la période de vacances, il reste une semaine de projet avant la soutenance et la remise finale. La séance du lundi a donc été consacrée à l’organisation du travail et à la répartition des tâches afin d’optimiser le temps restant et d’assurer l’aboutissement du projet.

1. Planification de la semaine de projet

Un point d’avancement global a été réalisé afin d’identifier les tâches restantes à mener avant la fin du projet. Les priorités ont été définies autour des éléments indispensables à la validation du projet, à savoir la réalisation matérielle du système, la documentation et la préparation de la soutenance.

2. Répartition des tâches

Le travail a été réparti entre les membres du groupe selon les compétences et l’avancement de chacun :

Soudure et assemblage du circuit imprimé : réalisation de la soudure des composants sur le PCB finalisé, suivie de tests de continuité et de mise sous tension progressive.

Rédaction des comptes rendus des dernières séances : finalisation et mise en forme des comptes rendus afin d’assurer la cohérence et la continuité du suivi du projet.

Création et organisation du dépôt GitHub : mise en place d’un dépôt GitHub destiné à centraliser le code, les fichiers de conception (schéma, PCB) et la documentation du projet.

Préparation du diaporama de soutenance : élaboration du support de présentation pour la soutenance, incluant la présentation du projet, de la démarche suivie, des résultats obtenus et des perspectives.

Conclusion

Cette séance a permis de structurer la dernière phase du projet tutoré. La planification et la répartition des tâches constituent une étape essentielle pour garantir la finalisation du système, la qualité des livrables et la réussite de la soutenance.

Compte rendu – Séance du mardi (dernière semaine du projet)

Contexte et objectifs de la séance

Cette séance du mardi s'inscrit dans la dernière semaine du projet tutoré. Les objectifs principaux étaient :

la poursuite des tâches de finalisation (GitHub et diaporama),

l'assemblage complet du système électronique,

la mise en service du montage et l'acquisition des premières données exploitables.

1. Avancement des livrables (GitHub et diaporama)

Une partie du groupe s'est chargée de la création et de l'organisation du dépôt GitHub du projet. Celui-ci permet de centraliser le code, les fichiers de conception électronique ainsi que la documentation associée.

En parallèle, d'autres membres du groupe ont poursuivi la préparation du diaporama de soutenance, en structurant la présentation autour des différentes étapes du projet, du travail réalisé et des résultats obtenus.

2. Problèmes rencontrés sur le montage électronique

Lors de l'assemblage et des tests du montage électronique, des dysfonctionnements ont été constatés au niveau du multiplexeur I²C. Le multiplexeur initialement utilisé s'est révélé défectueux, vraisemblablement en raison d'un problème de soudure ayant endommagé les lignes de communication (SDA/SCL). Il a donc été nécessaire de changer de multiplexeur afin de poursuivre les tests.

Malgré cette modification, le capteur spectral AS726x ne fonctionnait plus correctement et a été identifié comme hors service. Ce dysfonctionnement a empêché l'acquisition des données spectrales prévues initialement.

3. Montage complet et acquisition des données environnementales

Malgré les problèmes rencontrés sur le capteur spectral, le montage électronique complet a été réalisé et mis en fonctionnement avec les capteurs encore opérationnels.

Une phase d'acquisition de données a ainsi pu être menée sur :

la pression,

la température,

l'humidité,

en fonction du temps. Ces mesures ont permis de valider le fonctionnement global de la chaîne d'acquisition, de l'électronique jusqu'au traitement des données, et de disposer de résultats exploitables pour l'analyse et la présentation finale.

Conclusion

Cette séance du mardi a été marquée par des difficultés techniques liées à certains composants électroniques. Néanmoins, ces problèmes ont conduit à des choix techniques adaptés et à une réorientation partielle des tests. L'acquisition des données environnementales en fonction du temps a permis de valider une partie essentielle du système et de poursuivre le projet malgré les contraintes rencontrées.

Compte rendu – Séance du mercredi (dernière semaine du projet)

Contexte et objectifs de la séance

Lors de cette séance du mercredi, les objectifs étaient les suivants :

finaliser le montage électronique en insérant un nouveau capteur AS726x,

effectuer des mesures de longueurs d'ondes avec le capteur,

poursuivre les travaux sur Fusion 360 pour la conception du boîtier,

avancer sur la création du diaporama pour la soutenance du projet.

1. Montage électronique et insertion du nouveau AS726x

Le travail de montage a été poursuivi et nous avons pu insérer un nouveau capteur AS726x dans le montage. Après avoir reçu un remplacement pour le capteur défectueux, nous avons effectué les connexions nécessaires et vérifié la bonne intégration du composant dans la chaîne de mesure.

2. Mesures des longueurs d'ondes

Une fois le capteur AS726x fonctionnel, nous avons procédé à des mesures de longueurs d'ondes. Ces tests ont permis de valider la capacité du capteur à acquérir les données spectrales attendues et à les restituer de manière fiable.

Les résultats obtenus ont été analysés pour s'assurer de la cohérence des valeurs par rapport aux spécifications du capteur et aux attentes du projet.

3. Avancement sur Fusion 360

Le travail sur Fusion 360 a continué avec la révision et l'amélioration du modèle du boîtier. Nous avons peaufiné certains paramètres et ajusté les dimensions pour garantir une meilleure intégration des nouveaux composants (en particulier le capteur AS726x) et faciliter l'assemblage final du système.

4. Poursuite de la création du diaporama

Le diaporama de soutenance a également été poursuivi. La présentation a été structurée en différentes sections, couvrant :

l'introduction du projet,

les différentes étapes de développement (conception, montage, tests),

les résultats obtenus (mesures, analyse des données),

les perspectives et conclusions.

Les membres du groupe ont contribué activement à la rédaction du contenu et à la mise en forme des slides.

Conclusion

Cette séance du mercredi a permis d'effectuer des progrès importants sur plusieurs fronts :

la réintégration du capteur AS726x, permettant la reprise des mesures spectrales,

l'amélioration de la conception du boîtier sous Fusion 360,

l'avancement du diaporama, essentiel pour la soutenance de fin de projet.

Nous sommes désormais bien préparés pour la phase finale de tests et de présentation.

Compte rendu – Séance du jeudi (dernière semaine du projet)

Contexte et objectifs de la séance

Lors de cette séance du jeudi, plusieurs objectifs ont été atteints :

ajout d'un deuxième capteur AS726x pour comparer le NDVI entre la feuille et le ciel,

finalisation de l'intégration Arduino avec stockage des données sur la carte SD,

développement du programme Python pour l'acquisition, le traitement et la visualisation des données.

1. Ajout du deuxième capteur AS726x pour comparaison du NDVI

Un deuxième capteur AS726x a été ajouté au montage. Ce capteur a été positionné de manière à mesurer la réflectance du ciel, tandis que le premier capteur continue de mesurer celle de la feuille.

L'objectif est désormais de comparer les indices NDVI obtenus pour ces deux surfaces (ciel et feuille) afin de mieux comprendre l'impact de la végétation et des conditions ambiantes sur la réflexion de la lumière.

Les deux capteurs AS726x fonctionnent désormais simultanément, permettant de récolter les données nécessaires à la comparaison du NDVI en temps réel.

2. Finalisation de l'intégration Arduino et stockage sur carte SD

La partie Arduino a été finalisée en ce qui concerne le stockage des données. Les capteurs (y compris les AS726x, ainsi que les capteurs de température, pression et humidité) transmettent les données à l'Arduino, qui les enregistre sur une carte SD pour un stockage local.

Le programme Arduino a été modifié pour gérer l'enregistrement des données sous forme de fichiers CSV sur la carte SD, en prenant en compte la date et l'heure des mesures grâce à l'utilisation du module RTC.

3. Traitement et visualisation des données avec Python

Une fois les données enregistrées sur la carte SD, elles sont téléversées sur un programme Python pour le traitement et la visualisation. Le programme Python récupère les données du fichier CSV et trace plusieurs graphiques, notamment :

NDVI,

humidité,

pression,

température, tous en fonction du temps.

Les graphiques sont générés pour permettre une visualisation claire de l'évolution des paramètres mesurés, et facilitent l'analyse des variations du NDVI en relation avec les conditions environnementales (température, pression, humidité).

4. Avancement global du projet

Le projet avance bien, avec des progrès significatifs dans l'intégration des capteurs et la gestion des données. La comparaison du NDVI entre le ciel et la feuille nous permet d'obtenir des informations

précieuses sur l'état de la végétation, et les graphiques générés offrent une vue d'ensemble utile pour la présentation des résultats.

Conclusion

La séance du jeudi a permis d'atteindre plusieurs objectifs importants pour la fin du projet :

l'intégration complète du deuxième AS726x pour la comparaison du NDVI,

la finalisation de l'enregistrement des données sur la carte SD via l'Arduino,

la création d'un programme Python pour traiter et visualiser les données.

Le projet est désormais bien avancé et prêt pour la soutenance, avec des résultats expérimentaux solides et des outils de visualisation efficaces pour l'analyse des données.