Projet de merge des projets de l’A2

# Introduction

Le projet suivant a pour objectif de mettre à profit les différents projets mis au point durant l’année de A2.

Le projet sera constitué d’un Raspberry pour la station météo, d’un serveur linux pour stocker les datas, d’un backend en javascript (langage en cours d’acquisition en autodidacte), un front-end sur une page web et d’une application windows servant de dashboard en client lourd.

## Le MTH est de l’ordre de :

* Raspberry fonctionnel
* Serveur stockant les données collectées
* Back-end du serveur.
* Page web affichant les données récoltées.

## Le NTH est de l’ordre de :

* Client lourd en C#/CLI sur Windows d’affichage
* Logiciel Windows permettant de modifier les capteurs à faire fonctionner
* UG du logiciel permettant d’avoir des stats en direct (données capteurs, consommation électrique, etc…)
* UG du logiciel permettant de modifier l’affichage de la page web
* Création d’un système hermétique autonome (boitier, batterie)

# Prévisions temporelles

## Ordre de création

* Page web Dashboard
* Convertir le code de la station météo pour Raspberry
* Installation du serveur
* Back-end serveur

## Temps de création

* Dashboard 30h
  + Création de la page 10h
  + Création des scripts pour les graphiques 10h
  + Connexion de l’API 10h
* Raspberry 8h
  + Conversion du code 5h
  + Test du code et patch 3h
* Serveur 30h
  + Paramétrage réseau 10h
  + Installation de MySQL 10h
  + Création de la BDD 10h
* Back-end Rust – API ?h
  + Création de l’API
  + Connexion de la BDD à l’API

# Matériel nécessaire

## MTH

### Station météo :

* Raspberry pi 3B+
* Grove BME 280 – 23.6€
* Détecteur de lumière Grove V1.2 – 4.5€
* Grove Led v2.0 – 6.95€
* Module Grove Base Hat Zero 103030276 – 9.9€

Budget: 44.95

### Serveur BDD:

* Processeur 2c
* 4 go ram
* 500 ko/s ⬆️⬇️
* Volume de 10 Go de stockage
* Carte réseau wifi

# Code nécessaire

Raspberry : C++

Back-end : Rust

Portage Web : Html,CSS,JS

# Maquette

# Objectifs

## MTH :

Monter un système composé d’un raspberry servant de système embarqué, renvoyant des données vers une API, les reconduisant vers un serveur BDD en local. Un site hébergé sur ce même serveur appellerait l’API afin d’en récupérer les données et de les afficher sur un « dashboard » en ligne.

# Rendu

# Conclusion