Département de Génie Informatique

Etude d'efficacité des machines à café

Etudiants

Enes Sevim 19401842
Barış Kılıç 18401816

3 janvier 2024

Encadtants

Özgün Pınarer

Table des matières

1 Introduction

Le café est devenu un incontournable du XXIe siècle, surtout dans les environnements de travail. Dans les bureaux, les écoles et les lieux de travail, une quantité importante de café est consommée tout au long de la journée. Cependant, il est souvent difficile de prévoir la quantité de café qui sera consommée dans des espaces communs, et plus important encore, de déterminer la quantité qui sera consommée à une température agréable. Dans ce contexte, l'objectif de notre produit développé est d'optimiser le service de café offert, en particulier dans des espaces communs tels que les bureaux partagés, où de nombreuses personnes se réunissent. En anticipant la quantité de café à préparer, notre produit vise à améliorer l'expérience à la fois pour le consommateur et pour le prestataire de services. Si la quantité de café à préparer peut être prédite à l'avance, les buveurs pourront savourer leur café chaud, tout en réduisant le gaspillage de café qui ne sera pas consommé.

2 Matériaux Utilisés

Ce projet utilise divers matériaux pour mesurer l'efficacité de la machine à café. Chaque matériau a un rôle et une fonction spécifiques.

1. **NodeMCU ESP8266 :**
— *Objectif :* C'est le cerveau de notre circuit IoT. Il sert à recevoir les données lues par les capteurs et à effectuer des calculs sur ceux-ci et également à les envoyer à l'API. Il est également chargé d'informer l'API de l'état actif du système en envoyant des pings.
2. **Capteur de Poids HX711 :**
— *Objectif :* Mesurer la quantité de café consommée par les buveurs et déterminer la consommation totale.
3. **Capteur de Température LM35DZ :**
— *Objectif :* Mesurer la température du café et déterminer si elle se situe dans la plage de température buvable.
4. **LEDs :**
— *Objectif :* Utilisé pour fournir un retour visuel. Par exemple pour activer le système d'alarme lorsque le café devient froid.
5. **Résistances :**
— *Objectif :* Contrôler la résistance dans les circuits électriques et assurer le bon fonctionnement des capteurs.

Chaque matériau a été intégré dans ce projet avec un rôle, une fonction et un objectif spécifiques, permettant ainsi la collecte et l'analyse de données pour évaluer et optimiser l'efficacité de la machine à café. La combinaison de ces matériaux permet aux utilisateurs de tirer le meilleur parti de leur machine à café en facilitant la collecte et l'analyse de données.

3 Représentation schématique et mécanisme de circuit

Figure 1 explique comment fonctionne le côté NodeMCU du projet (capteurs, traitement des données, envoi à l'API). montre la représentation schématique de NodeMCU.

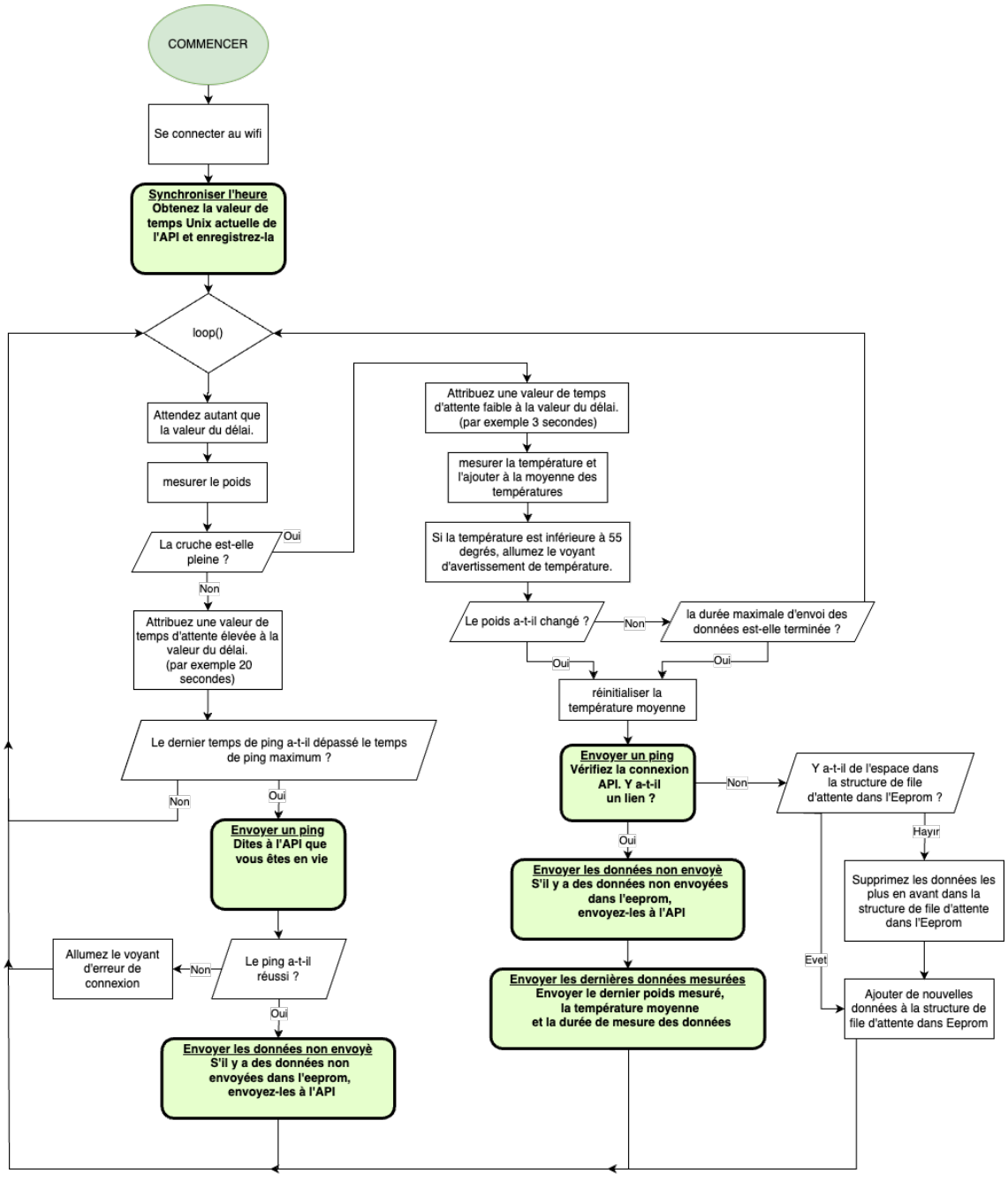

FIGURE 1 – Représentation schématique de NodeMCU

Dans notre projet IoT, nous commençons par établir une connexion réussie au réseau Wi-Fi. Ensuite, nous récupérons le temps actuel via l'API et enregistrons ce timestamp pour que nos envois de données soient associés dans le temps. Lorsque nous entrons dans la boucle principale, nous définissons un délai d'attente spécifique pour l'envoi des données,

ce qui permet à notre appareil de transmettre les données non pas en continu, mais à intervalles réguliers.

Nos capteurs de poids et de température effectuent leurs mesures respectivement. Si nos files d'attente de mesures sont pleines et qu'une nouvelle donnée est ajoutée, nous mettons à jour les anciennes données avec les nouvelles. Lorsque nous détectons un changement de poids, nous enregistrons ces nouvelles données dans notre système. Pour tester l'activité de notre connexion, nous envoyons un ping à l'API ; si nous rencontrons une erreur de connexion, nous allumons un voyant d'erreur pour signaler visuellement cette situation.

Si une erreur se produit lors d'un envoi et que nous avons des données non transmises, nous envoyons d'abord ces données. De plus, à chaque cycle, nous transmettons à l'API le poids le plus récent, la température moyenne et l'heure d'envoi des données. Pendant ce processus, si la température dépasse 55 degrés ou si notre temps maximal d'envoi de données est écoulé, notre appareil déclenche des protocoles spécifiques pour alerter. Pour éviter toute perte de données, nous conservons les données dans l'EEPROM dans une structure de file d'attente, améliorant ainsi la fiabilité de notre processus d'envoi de données.

Ce schéma de flux montre à quel point les processus de collecte et d'envoi de données de notre projet sont réguliers et automatisés. Dans notre rapport, nous détaillerons l'importance de cette régularité pour l'exactitude et la ponctualité des données.

4 Mécanisme de Backend

Notre API backend, conçue pour interagir avec un circuit NodeMCU, offre une gestion complète des données sensorielles et de l'état de vie d'un système. Grâce à des points d'extrémité spécifiques, notre API permet la réception et le stockage des données émanant du circuit NodeMCU, tout en fournissant une analyse visuelle dynamique desdites données via l'endpoint "/dashboard".

L'endpoint "/api/sensor" accepte les données de température, de poids et de temps Unix provenant du circuit NodeMCU, les convertit en un format exploitable, puis les enregistre dans une base de données MongoDB. Ceci garantit une gestion efficace des informations provenant du système sensoriel.

L'analyse visuelle est réalisée par l'intermédiaire de l'endpoint "/dashboard", qui récupère les données sensorielles stockées dans MongoDB et génère des graphiques temporels illustrant l'évolution des variables telles que la température et le poids. Ces graphiques fournissent un aperçu intuitif des tendances au fil du temps.

L'API comprend également des fonctionnalités de gestion de l'état de vie du système, avec les endpoints "/api/ping" et "/api/unix". L'endpoint "/api/ping" est utilisé pour mettre à jour l'état de vie du système à chaque ping reçu, tandis que "/api/unix" renvoie le temps Unix actuel, facilitant la synchronisation avec le circuit NodeMCU.

Notre système intègre également un mécanisme de planification asynchrone, permettant de mettre à jour automatiquement l'état de vie du système à intervalles réguliers.

En résumé, notre API backend offre une solution complète pour la collecte, le stockage

et l'analyse des données sensorielles, tout en assurant une gestion efficace de l'état de vie du système. Elle constitue un composant essentiel de notre infrastructure IoT, favorisant une interaction transparente entre les dispositifs physiques et l'interface utilisateur.

FIGURE 2 – Les données stockées dans MongoDB

5 Les graphiques pour l'analyse du système

Les graphiques d'analyse du système offrent une visualisation claire et concise des données collectées par le circuit NodeMCU, permettant une compréhension approfondie des performances du système. Nous présentons quatre graphiques distincts, chacun offrant un aperçu unique des variables clés.

Graphique : Temps vs Température

Ce graphique illustre l'évolution de la température au fil du temps. L'axe horizontal représente le temps, tandis que l'axe vertical affiche les valeurs de température. Les variations de température peuvent être facilement observées, permettant une identification rapide de tout comportement anormal. (Figure 3)

Graphique : Temps vs Poids

Ce graphique met en lumière la corrélation entre le temps et le poids du système. Les changements de poids sont tracés sur l'axe vertical par rapport au temps sur l'axe horizontal. Il offre une représentation visuelle des fluctuations de poids, permettant une analyse approfondie des tendances. (Figure 4)

Graphique : Températures vs Poids

Ce graphique compare directement les valeurs de température et de poids. Les températures sont affichées sur l'axe horizontal, les poids sur l'axe vertical. Cette visualisation

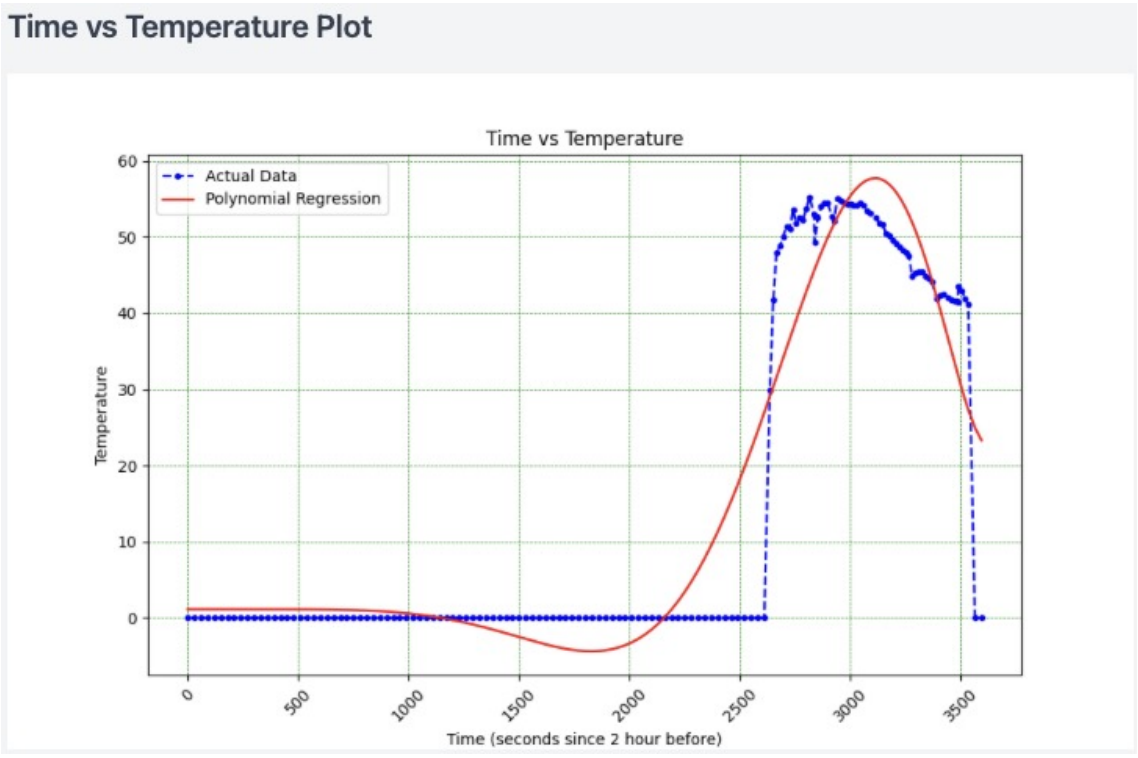

FIGURE 3 – Graphique : Temps vs Température

FIGURE 4 – Graphique : Temps vs Poids

permet de détecter des relations potentielles entre la température et le poids, facilitant une analyse plus approfondie des interactions entre ces deux variables. (Figure 5)

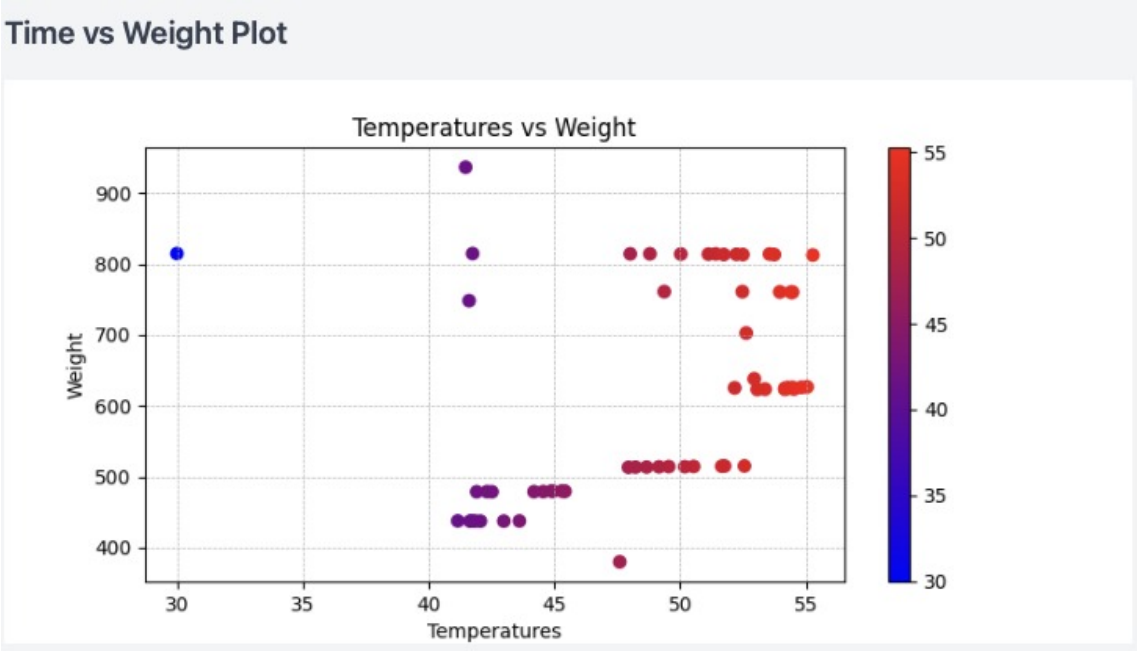

FIGURE 5 – Graphique : Température vs Poids

Graphique : Temps vs État de Vie

Le dernier graphique présente l'état de vie du système au fil du temps. L'axe horizontal représente le temps, tandis que l'axe vertical indique l'état de vie, avec des changements indiqués par des points discrets. Ce graphique permet de suivre l'évolution de l'état de vie du système, offrant un aperçu essentiel de son activité globale. (Figure 6)

Ces graphiques combinés fournissent une plateforme visuelle robuste pour l'analyse du système, permettant aux utilisateurs de prendre des décisions informées basées sur les tendances observées au fil du temps. Ils constituent un outil précieux pour la surveillance et l'optimisation continues du système basé sur les données réelles collectées.

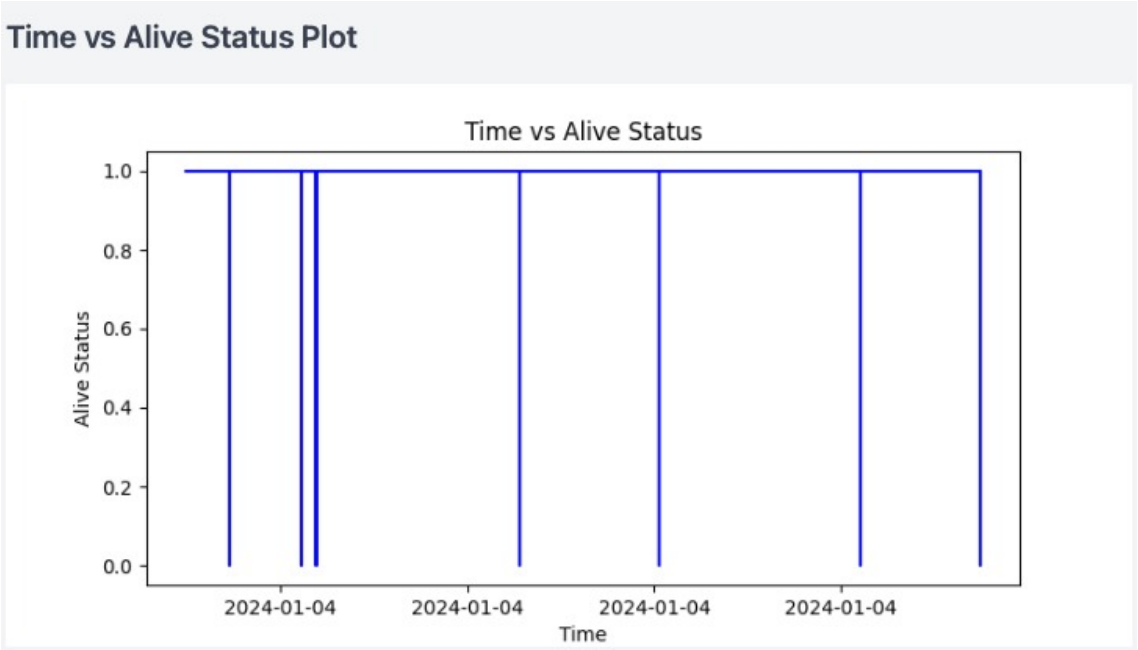

FIGURE 6 – Graphique : Temps vs État de Vie

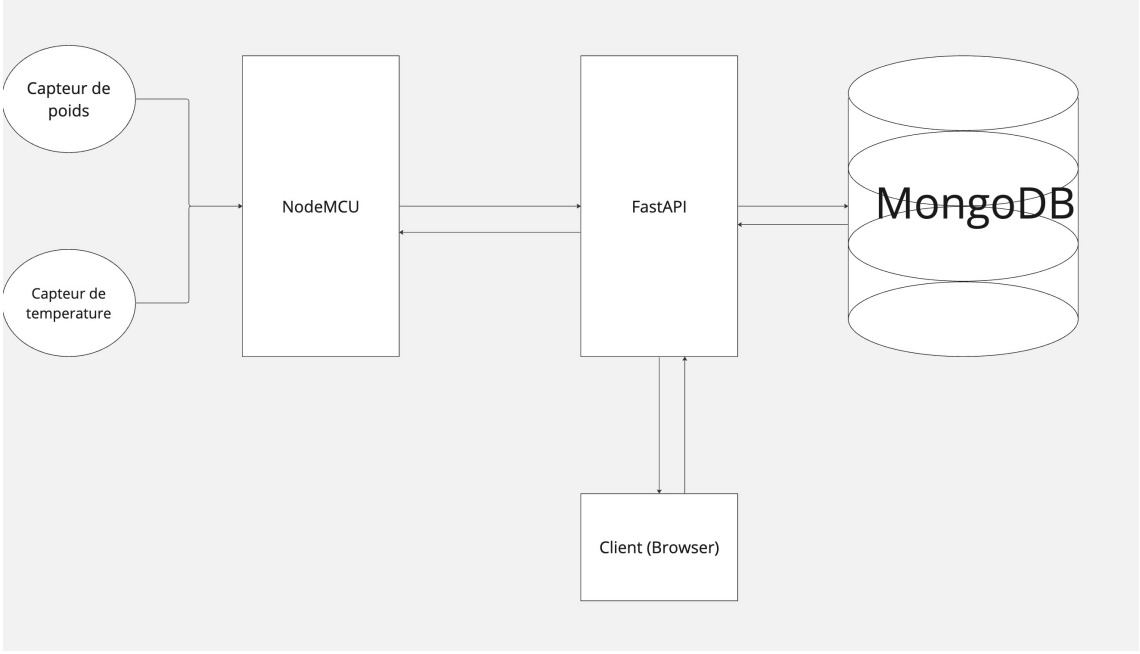

FIGURE 7 – Conception du système et mécanisme de travail

Références

[1] ESP8266EX Datasheet. (n.d.). Retrieved from `https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf`

[2] HX711 Datasheet. (n.d.). Retrieved from `https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf`

[3] LM35DZ Datasheet. (n.d.). Retrieved from `https://www.ti.com/lit/ds/symlink/lm35.pdf`