

# **RAPPORT DE STAGE**

BTS SIO – Option SLAM

Entreprise : IE-Products

Période : Du 01/01/2024 au 15/02/2024

Élève : Thirard Soulayman

Établissement : ORT Lyon

Tuteur entreprise : Souabni Badreddine

Directeur : Gharbi Kamel

Intitulé : Programmation d'appareils connectés  
(Sonde à température)

# SOMMAIRE

1. Remerciements	p3
2. Présentation de l'entreprise	p4
3. Contexte et objectifs du stage	p5
4. Missions réalisées	p6
5. Compétences développées	p10
6. Analyse personnelle et professionnelle	p11
7. Conclusion	p12
8. Annexes	p13

# Remerciements

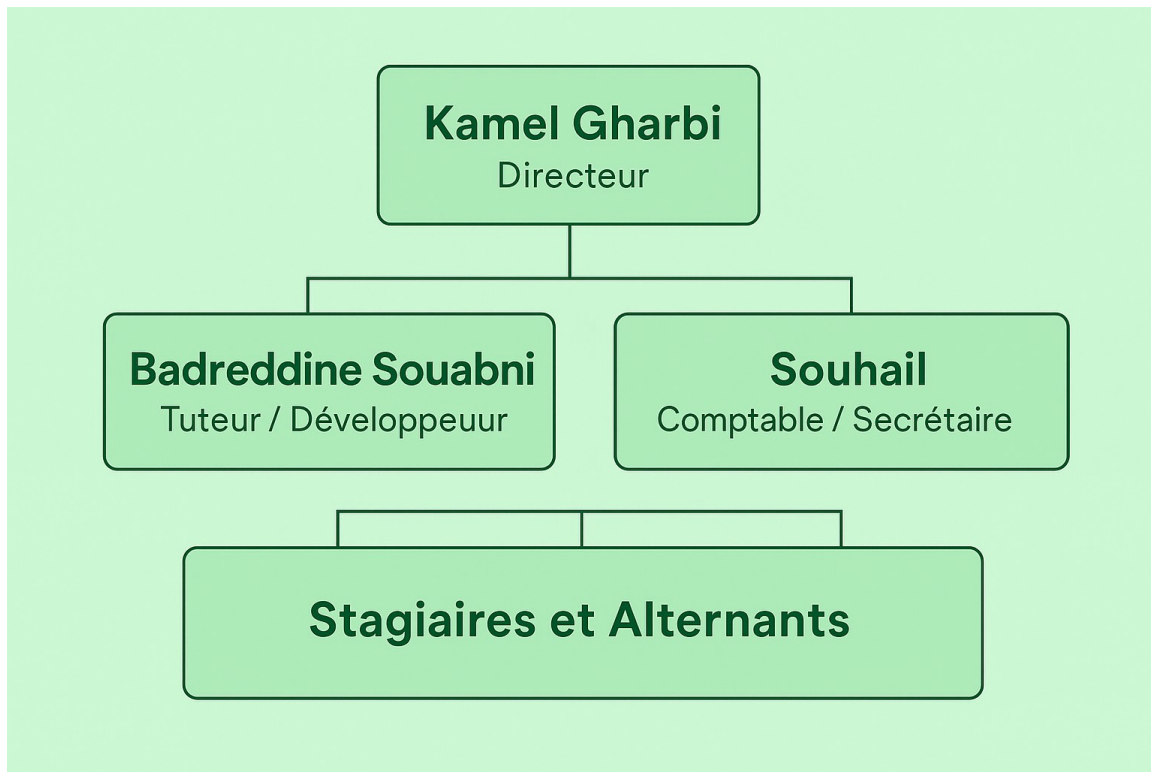
Je souhaite remercier toute l'équipe de **IE-Products** pour m'avoir à nouveau accueilli dans l'entreprise avec professionnalisme, bienveillance et confiance.

J'adresse mes remerciements les plus sincères à **Monsieur Souabni Badreddine**, mon tuteur de stage, pour son accompagnement technique, sa disponibilité constante, ainsi que pour ses conseils précieux tout au long de ce projet.

Je remercie également **Monsieur Gharbi Kamel**, directeur de l'entreprise, pour m'avoir renouvelé sa confiance, et pour m'avoir permis de participer à un nouveau projet concret au sein de l'équipe.

Enfin, je tiens à remercier l'équipe pédagogique du **CNED**, qui m'a permis de réaliser ce stage dans des conditions favorables à mon autonomie et à mon évolution professionnelle.

# Présentation de l'entreprise



**IE-Products** est une entreprise située à **Cusset, dans la métropole de Lyon**, spécialisée dans le **développement de solutions logicielles sur mesure**. Elle intervient aussi bien dans le secteur **B2B (entreprises)** que **B2C (grand public)**, en concevant des **applications mobiles et web** adaptées aux besoins de ses clients.

L'entreprise se distingue par sa **capacité à innover** et à **s'adapter rapidement aux évolutions technologiques**. Elle intègre dans ses projets des outils récents et des approches modernes, favorisant ainsi la performance, la fluidité et la compatibilité multi-plateforme.

IE-Products travaille actuellement à la relance de ses activités informatiques, en mettant l'accent sur des **produits numériques utiles, accessibles et intuitifs**. Elle propose aussi des solutions hybrides (app + support physique), dans un esprit de polyvalence et de complémentarité.

## Contexte et objectifs du stage

Dans le cadre de ce second stage chez **IE-Products**, l'objectif principal était de participer à la mise en place d'une **solution connectée dédiée au secteur agroalimentaire**, et plus précisément aux professionnels comme les **bouchers et charcutiers**. L'entreprise souhaitait développer un **système intelligent capable de mesurer, surveiller et contrôler la température** de chambres froides ou divers outils comme une sonde à distance, à travers une interface connectée.

Ce projet s'inscrit dans une **stratégie B2B** innovante visant à proposer aux commerçants un **outil numérique simple, efficace et sécurisé**, pouvant interagir avec une **application mobile**. L'idée étant, par exemple, de permettre à un artisan d'accéder à tout moment, depuis son smartphone, à l'état thermique de ses équipements de conservation, avec des fonctionnalités de notification, d'historique, voire de contrôle à distance.

Ma mission consistait à concevoir **une application back-end** assurant la **liaison entre un microcontrôleur Raspberry Pi Pico**, une **sonde de température**, et la **base de données du serveur**. J'ai dû **programmer une API spécifique**, gérer les échanges de données, et veiller à la stabilité du système dans un environnement réel.

Ce stage m'a confronté à un **nouveau type d'environnement technique**, incluant à la fois des compétences en **programmation bas niveau**, en **électronique (soudure, manipulation de composants)**, et en **architecture de communication réseau**. Il représentait un véritable défi d'apprentissage et d'adaptation.

# Missions Réalisées

## 1. Autoformation à l'environnement Raspberry Pico et MicroPython

Au cours des deux premières semaines de mon stage, j'ai entamé une phase d'**autoformation intensive** visant à me familiariser avec le **Raspberry Pi Pico**, un microcontrôleur que je n'avais jamais utilisé auparavant, ainsi qu'avec le langage **MicroPython**, indispensable pour la suite du projet.

Cette mission avait pour but de me rendre autonome dans l'utilisation de ce type de matériel afin de pouvoir ensuite concevoir une application back-end en lien avec une sonde de température. J'ai pour cela utilisé les **documentations officielles** du Raspberry Pi et de MicroPython, ainsi que des tutoriels spécialisés pour débiter sur le sujet.

J'ai tout d'abord appris à **connecter le Raspberry Pico en USB à mon poste de travail**, à installer le firmware MicroPython et à **communiquer avec lui via un interpréteur série**. J'ai ensuite expérimenté l'envoi d'instructions simples pour comprendre la gestion des **ports GPIO**, en m'entraînant notamment à **allumer et éteindre la LED embarquée**.

L'un des défis principaux était de m'adapter rapidement à un environnement technique totalement nouveau, qui mélangeait **programmation bas niveau, manipulation physique** (connexion de câbles, soudure de pins) et **notions d'électronique embarquée**. Grâce à une documentation claire et à une pratique quotidienne, j'ai progressivement acquis les bases nécessaires.

En fin de mission, j'étais capable de contrôler le comportement du microcontrôleur, et même de **transmettre des données via une extension de stockage micro-SD**, ce qui représentait une première avancée concrète vers la mise en place du système complet.

*(Annexe 1 : documentation utilisée, photos de l'installation et du test LED.)*

## **2. Préparation physique du matériel : soudure des pins sur le Raspberry Pico**

Lors de la troisième semaine de stage, j'ai été chargé d'une mission plus matérielle et manuelle : **souder les broches (ou "pins") sur le Raspberry Pi Pico**. Cette étape était indispensable pour permettre la **connexion future de sondes externes**, notamment un ampèremètre, ainsi que la **liaison avec une sonde de température** prévue pour les tests ultérieurs.

Pour cela, j'ai utilisé un **fer à souder de précision**, de l'**étain** adapté à l'électronique, et manipulé les composants avec précaution. La mission consistait à réaliser une **soudure propre et fiable** afin de garantir la conductivité électrique sans risquer d'endommager le microcontrôleur.

La principale difficulté résidait dans le **manque d'expérience** avec ce type de tâche manuelle, qui demande une certaine **dextérité, précision et patience**. Chaque soudure devait être suffisamment robuste tout en étant fine, sans court-circuit ou dépôt excessif d'étain.

Grâce à une prise en main progressive, plusieurs essais sur des composants de test, et en respectant les consignes de sécurité, j'ai finalement réussi à souder correctement les pins du Raspberry Pico. Cette étape a constitué une **base essentielle pour la suite du projet**, car elle permettait d'établir les connexions avec les capteurs et de valider les circuits physiques à tester.

*(Annexe 2 : photographies de la soudure sur le Raspberry Pi Pico)*

### 3. Développement du back-end : API de communication avec le Raspberry Pi Pico

La troisième et dernière mission de mon stage a consisté à concevoir et développer une application back-end sous forme d'une API REST. Cette API avait pour objectif principal de recevoir et traiter les données de température mesurées par une sonde reliée à un Raspberry Pi Pico, dans un contexte d'application IoT destiné au secteur agroalimentaire.

#### *Objectif*

L'objectif était de fournir une interface de communication serveur capable d'exposer des points d'accès (endpoints) permettant à des applications clientes (comme une application mobile) de récupérer des mesures de température en temps réel, ou de les stocker pour exploitation ultérieure.

#### *Technologies utilisées*

- Langage : Python
- Framework : Flask (micro-framework web léger et rapide)
- Outils de test : Postman, pour tester les routes de l'API et simuler des requêtes HTTP
- Matériel : Raspberry Pi Pico, sonde de température

#### *Travaux réalisés*

J'ai structuré un projet Flask avec :

- J'ai utilisé l'api de google météo pour m'entraîner
- des routes REST sécurisées par méthode HTTP (GET/POST),
- une connexion avec une base de données SQLite locale (ou un fichier JSON temporaire selon les phases de test),
- des tests unitaires avec Postman pour simuler l'envoi de valeurs captées par la sonde (via le port série ou en émulation).

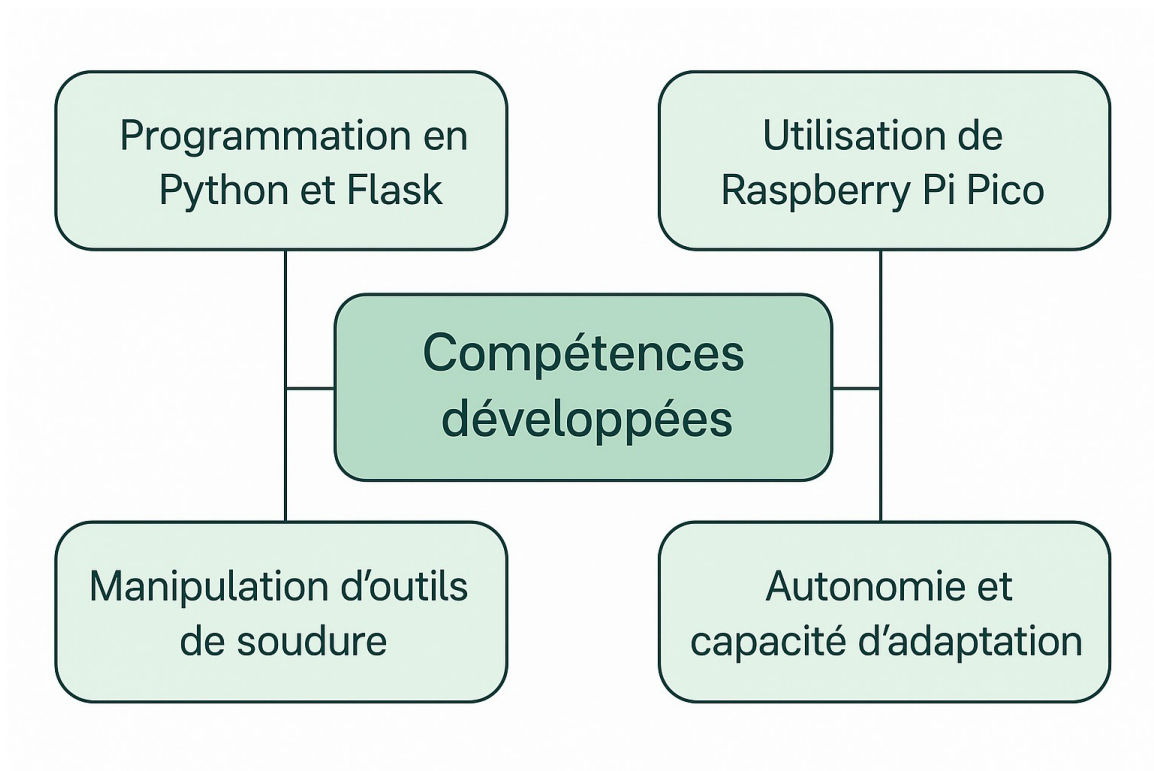


Par manque de temps, l'interface utilisateur (front-end) n'a pas pu être réalisée. Toutefois, l'API est pleinement fonctionnelle et prête à être intégrée dans une future application mobile ou web.

Cette mission m'a permis de renforcer mes compétences en développement back-end, en conception d'API, et en interaction avec du matériel embarqué dans un contexte professionnel concret.

**(Annexe 3 :** extraits de code avec Flask.)

## Schéma des compétences développées



# Analyse Personnelle et Professionnelle

Ce second stage m'a confronté à un environnement de travail très différent de mon premier projet. Là où j'étais auparavant en terrain relativement connu (développement mobile avec Flutter), j'ai cette fois dû sortir de ma zone de confort pour aborder des domaines entièrement nouveaux pour moi, à commencer par le **matériel embarqué**, l'**électronique** et la **programmation à bas niveau**.

Sur le plan **professionnel**, ce stage m'a permis de prendre conscience de la diversité des situations qu'un développeur peut rencontrer. J'ai découvert l'importance de comprendre non seulement le logiciel, mais aussi le matériel, et la manière dont les deux interagissent. J'ai appris à manipuler un microcontrôleur, à souder des composants, à lire des schémas techniques, à interpréter des signaux électroniques, et à faire communiquer des éléments physiques avec une interface logicielle.

J'ai aussi pu consolider mes bases en **développement Python** dans un cadre plus orienté production, en concevant une **API REST** avec Flask répondant à un besoin métier réel. La rigueur de conception, la clarté du code, la documentation et les tests sont autant de compétences que j'ai approfondies. La contrainte de devoir intégrer mon code dans une architecture existante m'a appris à mieux structurer mes projets et à anticiper leur évolutivité.

Du point de vue **personnel**, cette expérience m'a réellement challengé. Il m'a fallu beaucoup d'adaptabilité pour assimiler rapidement des notions techniques complexes que je n'avais jamais abordées. J'ai appris à m'organiser seul, à chercher efficacement dans la documentation, à me tromper, à corriger, et à progresser étape par étape. J'ai gagné en **autonomie**, en **esprit d'analyse**, mais aussi en **confiance** dans ma capacité à aborder des problématiques techniques nouvelles.

Cette expérience m'a également montré que je suis capable de m'impliquer sérieusement dans un projet concret, même lorsque celui-ci dépasse mes compétences initiales. Elle a renforcé mon attrait pour les projets **technologiques hybrides**, mêlant logique métier, interactions physiques, et innovation au service d'un besoin réel.

## Conclusion

Ce second stage au sein de **IE-Products** m'a offert une opportunité concrète de sortir des cadres traditionnels du développement logiciel pour explorer un projet mêlant **matériel embarqué, électronique et développement back-end**. Ce contexte technique exigeant m'a permis d'enrichir mes compétences tout en me confrontant à des défis concrets, proches des réalités d'un projet professionnel.

Au-delà des compétences techniques que j'ai pu acquérir — qu'il s'agisse de la manipulation d'un microcontrôleur, de la soudure de composants ou du développement d'une API REST —, ce stage m'a surtout appris à **faire face à l'inconnu**, à structurer ma démarche d'apprentissage et à m'adapter à des situations complexes et inédites.

Je ressors de cette expérience avec une vision plus large du métier de développeur, conscient qu'il ne se limite pas à un langage ou à un outil, mais qu'il s'inscrit dans un **écosystème global** où la communication entre les composants, les machines et les utilisateurs est essentielle.

Ce stage m'a aussi conforté dans mon envie de **continuer à apprendre, à expérimenter et à créer** des solutions utiles, robustes et innovantes. Il a représenté une étape marquante dans mon parcours de formation, en me permettant de passer de l'acquisition de connaissances à leur mise en œuvre concrète, dans un cadre professionnel exigeant.

En résumé, ce stage m'a apporté bien plus qu'une simple mise en pratique de mes acquis : il m'a permis de me projeter dans un rôle de développeur polyvalent, capable d'apprendre rapidement, de s'adapter, de collaborer et de produire du code utile dans des contextes variés.

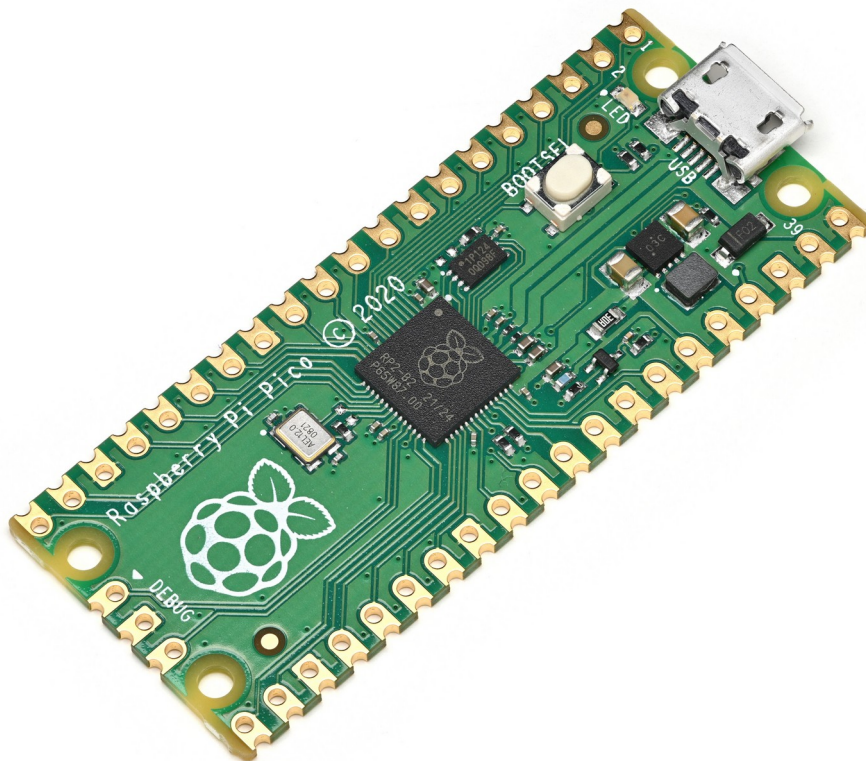
# Annexes

## Annexe 1

*Liens vers la documentation de micro-python et du Raspberry Pi Pico :*

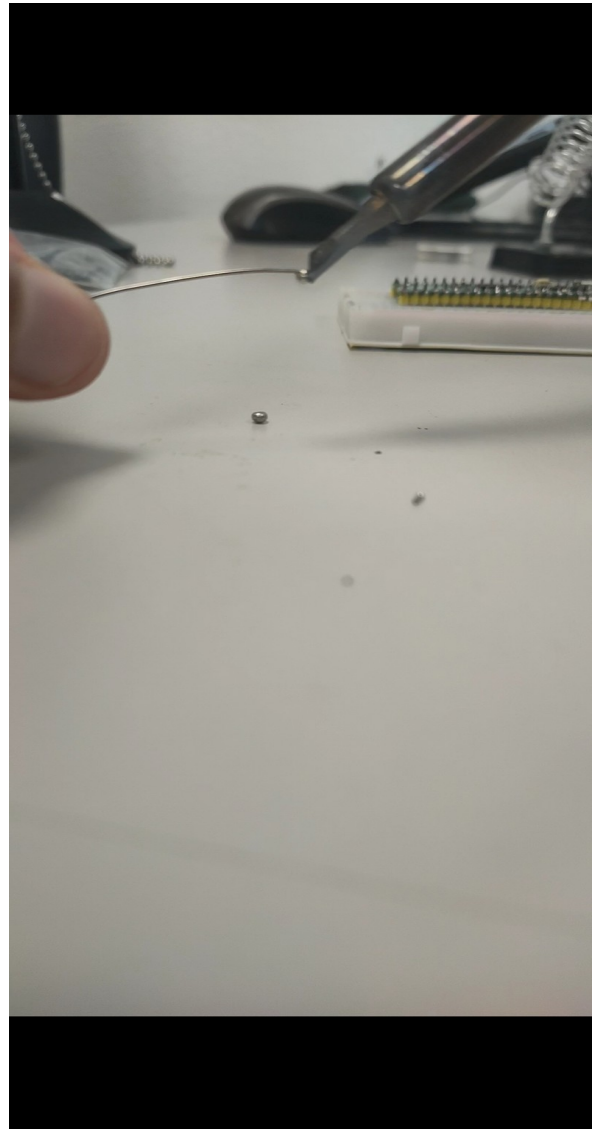
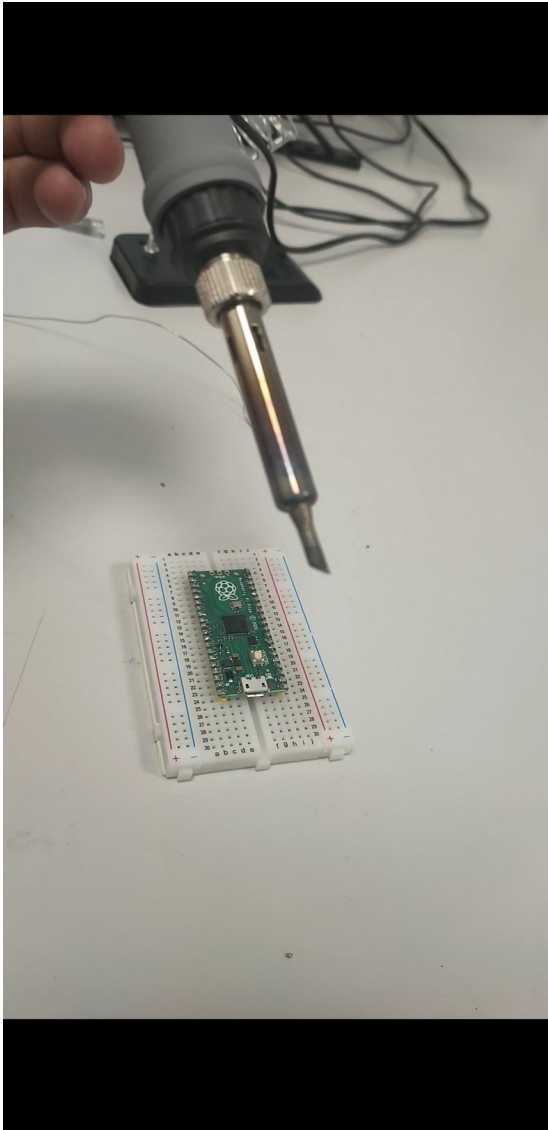
*<https://micropython.org/> et <https://www.raspberrypi.com/documentation/>*

*Image d'un Raspberry Pi Pico :*



## Annexe 2

*Soudure avec un fer à chauffer :*



## Annexe 3

Captures d'écran du code : **database.py** avec opérations CRUD

```
database.py > ...
1  import sqlite3
2
3  def init_db():
4      conn = sqlite3.connect('data.db')
5      cursor = conn.cursor()
6      cursor.execute('''
7          CREATE TABLE IF NOT EXISTS temperature_logs (
8              id INTEGER PRIMARY KEY AUTOINCREMENT,
9              value REAL NOT NULL,
10             timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
11         )
12     ''')
13     conn.commit()
14     conn.close()
15
16 def insert_temperature(value):
17     conn = sqlite3.connect('data.db')
18     cursor = conn.cursor()
19     cursor.execute('INSERT INTO temperature_logs (value) VALUES (?)', (value,))
20     conn.commit()
21     conn.close()
22
23 def get_all_temperatures():
24     conn = sqlite3.connect('data.db')
25     cursor = conn.cursor()
26     cursor.execute('SELECT * FROM temperature_logs ORDER BY timestamp DESC')
27     data = cursor.fetchall()
28     conn.close()
29     return data
30
31 def update_temperature(id, new_value):
32     conn = sqlite3.connect('data.db')
33     cursor = conn.cursor()
34     cursor.execute('UPDATE temperature_logs SET value = ? WHERE id = ?', (new_value, id))
35     conn.commit()
36     conn.close()
37
38 def delete_temperature(id):
39     conn = sqlite3.connect('data.db')
40     cursor = conn.cursor()
41     cursor.execute('DELETE FROM temperature_logs WHERE id = ?', (id,))
42     conn.commit()
43     conn.close()
44
```



*Fichier micro-python du Raspberry Pi Pico + app.py :*

```
rasp.py > ...
1  import urequests
2  import machine
3  import time
4
5  # Simule une lecture de température (ex: TMP36 branchée à GP26)
6  sensor = machine.ADC(26)
7
8  def get_temperature():
9      value = sensor.read_u16()
10     voltage = (value / 65535.0) * 3.3
11     temperature_c = (voltage - 0.5) * 100
12     return round(temperature_c, 2)
13
14  while True:
15     temp = get_temperature()
16     try:
17         r = urequests.post("http://<IP_SERVEUR>:5000/api/temperature", json={"temperature": temp})
18         print(r.text)
19     except Exception as e:
20         print("Erreur d'envoi :", e)
21     time.sleep(60) # envoie toutes les 60 secondes
22
```

```
1  from flask import Flask, request, jsonify
2  from flask_cors import CORS
3  from database import init_db, insert_temperature, get_all_temperatures
4
5  app = Flask(__name__)
6  CORS(app)
7
8  # Initialisation de la base
9  init_db()
10
11  @app.route('/')
12  def home():
13     return "API de monitoring de température (Raspberry Pi Pico)"
14
15  @app.route('/api/temperature', methods=['POST'])
16  def post_temperature():
17     data = request.get_json()
18     temp = data.get("temperature")
19     if temp is None:
20         return jsonify({"error": "Champ 'temperature' requis"}), 400
21
22     insert_temperature(temp)
23     return jsonify({"message": "Température enregistrée avec succès", "value": temp}), 201
24
25  @app.route('/api/temperature', methods=['GET'])
26  def get_temperature():
27     data = get_all_temperatures()
28     result = [{"id": row[0], "value": row[1], "timestamp": row[2]} for row in data]
29     return jsonify(result)
30
31  if __name__ == '__main__':
32     app.run(debug=True, host='0.0.0.0', port=5000)
33
```