

Institut National de Radioélectricité et de Cinématographie

Enseignement technique secondaire de qualification

Accès aux études supérieures



Avenue Jupiter, 188

1190 Forest

NutriTech

(Pompe d'alimentation entérale)

Projet personnel de Apio Makani Julianna

Pour l'obtention du certificat de qualification

Technicien(ne) en informatique

Année scolaire : 2025 – 2026

## **Remerciement**

Avant toute chose, j'aimerais adresser mes remerciements aux personnes qui m'ont aidé tout au long du projet.

D'abord mes professeurs d'informatique qui ont accepté mon projet. M. Kajjoua, M. Mdiker et M. Jannah qui m'ont donné des conseils.

Pour terminer, je voudrais remercier ma mère qui m'a encouragé dans ce projet. Elle a été un soutien tout du long et permit d'avancer dans projet. J'aimerai aussi remercier mon grand frère. Grace à lui, j'ai pu avoir l'idée de cette pompe d'alimentation entérale.

## Table des matières

1. Introduction.....	5
2. Cahier des charges.....	6
2.1 Cahier des charges du matériels .....	6
2.2 Cahier des charges des logiciels .....	9
3. Matériel utilisé .....	10
3.1 Raspberry pi 4B .....	10
3.1.1 Fonctionnement et choix dans le projet.....	10
3.1 Pompe péristaltique avec moteur pas-à-pas.....	11
3.1.1 Moteur pas-à-pas .....	12
3.1.2 Fonctionnement et choix dans mon projet.....	13
3.1.3 Problème rencontré :.....	13
3.2 Débitmètre .....	14
3.2.1 Fonctionnement et choix dans le projet.....	14
3.3 Ecran LCD .....	16
3.3.1 Fonctionnement et choix dans le projet.....	16
3.4 Keypad.....	17
3.4.1 Fonctionnement et choix dans le projet.....	17
3.5 Bouton poussoir .....	18
3.5.1 Fonctionnement et choix dans le projet.....	18
3.6 Buzzer piézoélectrique.....	19
3.6.1 Fonctionnement et choix dans le projet.....	19
3.7 Alimentation 12v.....	20
4. Logiciel et technologie.....	21
4.1 Python .....	21
4.2 Flask.....	21
4.2.1 Fonctionnement dans le projet: .....	22

4.3	MQTT .....	23
4.3.1	Mosquitto .....	24
4.3.2	Fonctionnement dans mon projet .....	24
4.4	SQLite .....	25
4.4.1	Choix .....	25
4.4.2	Fonctionnement dans le projet : .....	26
4.5	Tailscale .....	28
4.5.1	Fonctionnement et choix dans mon projet.....	28
4.6	Nginx.....	29
4.6.1	Fonctionnement dans mon projet .....	29
4.7	Cloudflare.....	30
4.6.1	Cloudflare Tunnel .....	30
4.6.2	Fonctionnement et choix dans le projet.....	31
4.6.3	Problèmes et solutions rencontrés : .....	32
4	Architecture du système .....	34
5.1	Topologie Cloudflare .....	34
5.2	Schéma électronique .....	35
5.3	Schéma mécanique .....	36
5.	Conclusion.....	37
6.	Bibliographie.....	38
7.	Annexe .....	40

## 1. Introduction

La problématique de la vie nous surprend parfois jusqu'à la changer. Certaines personnes se retrouvent avec des situations d'handicap, qui nécessite des adaptations particulières comme c'est le cas pour mon grand frère. Ce dernier est alimenté de façon entérale (c'est-à-dire directement via l'estomac) avec une sonde gastrique.

Le problème des systèmes mis à disposition à l'heure actuelle est qu'ils ne sont pas complètement autonomes. Il faut l'allumer et l'éteindre sur un bouton physique situé sur la pompe, sinon un bip très aigu reste tant que personne n'appuie dessus. Mon frère, qui est dans l'incapacité de réaliser cette action, doit supporter le bruit. S'il doit manger avec cette machine, une personne doit être présente, doit se déplacer. Ce problème nous contraint à ne pas vraiment utiliser le produit à cause de ce défaut.

Cette dernière année en secondaire, il nous est demandé de réaliser un TFE avec un contexte d'implémentation. J'ai donc décidé de réaliser cette pompe de manière à ce qu'elle réponde à mon besoin. L'objectif de ce TFE est de réaliser cette pompe d'alimentation entérale et de pouvoir la contrôler avec une interface Web.

## 2. Cahier des charges

### 2.1 Cahier des charges du matériels

2. Numéro	Matériels	Explication	Quantité	Liens
1	Pompe péristaltique avec moteur pas à pas	Moteur pas-à-pas qui fait tourner une pompe avec un tuyau à l'intérieur.	1	<a href="#">Pompe péristaltique avec moteur pas à pas, pompe à tuyau de dosage à grand débit Micro 12-24v Roller : Amazon.fr: Commerce, Industrie et Science</a>
2	Ecran LCD	Affiche les informations voulus.	1	<a href="#">Freenove I2C IIC LCD 1602 Module, New Type TWI Serial 16x2 Display, Compatible with Arduino Raspberry Pi Pico ESP32 ESP8266 : Amazon.fr: Commerce, Industrie et Science</a>
3	Keypad	Pavé qui sert à changer les informations physiquement.	1	<a href="#">Clavier Matrix Array pour Ardu37, 16 touches, clavier à membrane, 3x4, 4x4, 1-10 pièces - AliExpress</a>

4	Flowsensor (débitmètre)	Permet de mesurer le débit écoulé par la pompe dans le tuyau.	1	<a href="#">Capteur de débit d'eau pour Ardu37, effet de contrôle du concentrateur, hall 1-30L/min, 1/2 en effet, 1PC</a>
5	Buzzer piézo	Emet un effet sonore.	1	<a href="#">Module de capteur d'alarme de sonnerie Active 3pin de KY-012 pour Kit de démarrage arduino bricolage KY012 - AliExpress</a>
6	Bouton poussoir	Start/stop	1	<a href="#">DAOKAI 100PCS Tactile Tact Bouton Poussoir 4Pin 6x6x5mm Interrupteur Momentané Vertical Bouton Miniature Composants Électroniques pour Panneau PCB : Amazon.fr: Commerce, Industrie et Science</a>
7	A4988	Driver pour le moteur pas-à-pas	1	<a href="#">WJMY A4988 Stepper Motor Driver Module</a>

				<a href="#">5pcs Module Pilote de Moteur Pas a Pas avec Radiateur pour Imprimante 3D RepRap :</a> <a href="#">Amazon.fr: Commerce, Industrie et Science</a>
8	Alimentaion 12v 2A	Alimentation pour le moteur pas-à-pas	1	<a href="#">Adaptateur Secteur 12V 2A 24W Universel avec Câble de 1,5m, Alimentation AC-DC pour Haut-Parleurs, Caméras CCTV, Routeurs et Appareils Électroniques, Transformateur de Haute Qualité 5,5mm x 2,5mm :</a> <a href="#">Amazon.fr: High-Tech</a>

## 2.2 Cahier des charges des logiciels

Numéro	Logiciels	Explication	Prix
1	Flask	Framwork web	0€
2	Tailscale	VPN	0€
3	SQLite	Base de données local	0€
4	Cloudflare		0€
5	MQTT		0€
6	Mosquitto	Broker MQTT	0€
7	Autodesk Fusion 360	Conception 3d	Version gratuite

## 3. Matériel utilisé

### 3.1 Raspberry pi 4B

Un Raspberry Pi est un mini-ordinateur puissant et compact, de la taille d'une carte bancaire, conçu pour apprendre l'informatique et réaliser des projets électroniques ou de robotique. Il fonctionne comme un véritable ordinateur : il possède un processeur, de la mémoire vive (jusqu'à plusieurs Go de RAM), des ports USB, une



sortie HDMI pour écran, un port Ethernet pour Internet et du Wi-Fi intégré. Il peut exécuter un système d'exploitation comme Raspberry Pi OS et permettre d'utiliser des logiciels, programmer en Python, ou encore créer des serveurs.

En plus de ses fonctions de simple ordinateur, il dispose de broches appelées GPIO, qui permettent de connecter des capteurs, des moteurs ou d'autres composants électroniques. Cela le rend très utilisé dans les projets de domotique, de robotique et d'automatisation. Le Raspberry Pi 4 est apprécié car il est peu cher, polyvalent et assez puissant pour son format, ce qui en fait un outil très populaire.

#### 3.1.1 Fonctionnement et choix dans le projet

Le Raspberry pi a été retenu pour ce le projet de pompe d'alimentation en raison d'une utilisation avec un autre projet déjà réalisé. D'autre part, il remplit plusieurs fonctions comme faire à la fois serveur et contrôleur pour les composants en un mini-ordinateur.

En premier lieu le fait qu'il soit un mini-ordinateur facilite la tâche. Une interface doit pouvoir interagir avec le dispositif. Dans cette optique, le Raspberry pi est capable de le faire sans ajouter un serveur en plus.

A cela s'ajoute que le mini-ordinateur est capable de contrôler des composants électroniques grâce au GPIO, ce qui permet de piloter divers capteurs et autres périphériques.

### 3.1 Pompe péristaltique avec moteur pas-à-pas

Une pompe péristaltique est un type de pompe qui permet de déplacer un liquide en écrasant un tube flexible à l'aide de rouleaux ou de galets qui tournent. En avançant, ces rouleaux écrasent le tuyau et poussent le liquide vers l'avant, un peu comme si on faisait avancer le fluide par vagues successives. Lorsque la pression se relâche, le tube reprend sa forme initiale et aspire automatiquement du liquide derrière lui. Ce système a l'avantage d'être très propre, car le liquide ne touche que l'intérieur du tuyau, ce qui le rend utile dans des domaines comme la médecine, les laboratoires ou certains systèmes industriels. En revanche, le débit est généralement limité et le tuyau peut s'user avec le temps à cause des compressions répétées.



### 3.1.1 Moteur pas-à-pas

Un moteur pas-à-pas est un moteur électrique qui tourne par petits mouvements réguliers appelés “pas”. Contrairement à un moteur classique qui tourne en continu, celui-ci avance d’un angle précis à chaque impulsion électrique qu’il reçoit. Cela permet de contrôler très précisément sa position, sa vitesse et le nombre de tours effectués. Il est souvent utilisé dans les imprimantes 3D, les robots ou les machines où il faut un mouvement précis. Son avantage principal est qu’il n’a pas forcément besoin de capteur pour savoir où il se trouve, mais il peut parfois perdre sa précision s’il est trop chargé ou mal utilisé.



Pour piloter le moteur, il faut un driver comme la A4988.

Il sert à contrôler un moteur pas-à-pas en envoyant des impulsions électriques précises. Grâce à lui, on peut régler la vitesse, le sens de rotation et le nombre de pas du moteur. Il permet aussi de gérer le courant envoyé au moteur pour éviter qu’il ne surchauffe ou ne soit endommagé.

Le A4988 est souvent utilisé avec des cartes comme Arduino ou Raspberry Pi. Il est apprécié car il est simple à utiliser, peu coûteux et capable de gérer des moteurs avec une bonne précision.

### **3.1.2 Fonctionnement et choix dans mon projet**

L'utilisation d'une pompe péristaltique avec moteur pas-à-pas a été choisie en fonction de différents facteurs. Comme expliqué précédemment, l'avantage de la pompe péristaltique plutôt qu'une pompe dite classique est que, grâce à son tube flexible intégré, le liquide traversant le tube reste à l'intérieur. Avec des dispositifs médicaux, comme une alimentation entérale, la propreté du système est primordiale.

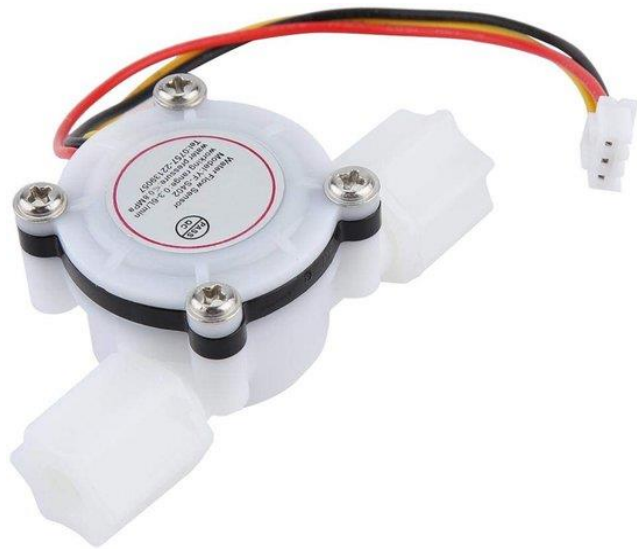
Dans un second temps, l'ajout du moteur pas-à-pas à la pompe est un élément non négligeable. Grâce à ce moteur, le débit du liquide poussé peut être contrôlé de façon très précise. La gestion des « pas » permet ainsi de le gérer avec plus de précision et constance. Avec les pompes sans ce moteur, on a tendance à avoir une précision moins bonne. En effet, à chaque lancement, elle peut avoir un débit et la vitesse varient et ne sont pas souvent les mêmes malgré qu'il n'y ait eu aucun changement. Le moteur pas-à-pas vient régler ces défauts. Dans une pompe d'alimentation, la constance est très importante.

### **3.1.3 Problème rencontré :**

Avec le driver, il y a eu plusieurs soucis. Sur le driver, il s'y trouve un potentiomètre. Il faut le régler sur la bonne tension pour qu'il y est assez de courant. Malgré le courant nécessaire, le moteur n'arrivait pas à tourner.

## 3.2 Débitmètre

Un débitmètre YF-S402C est un capteur qui permet de mesurer le débit d'un liquide, généralement de l'eau, dans un tuyau. Le capteur fonctionne grâce à une turbine qui tourne à l'intérieur lorsque qu'un liquide s'écoule à l'intérieur la turbine fait un tour complet, ça crée un signal électrique qui est envoyé au Raspberry pi. Ceci est appelé un



pulse. Plus le liquide circule rapidement, plus il y a de pulse. En comptant ces impulsions, un microcontrôleur peut calculer la quantité d'eau qui passe, par exemple en litres par minute. Ce type de débitmètre est souvent utilisé dans des projets électroniques et des systèmes automatisés pour surveiller ou contrôler la consommation de liquide.

### 3.2.1 Fonctionnement et choix dans le projet

Le but du débitmètre dans le projet est de récupérer le débit du liquide poussé par la pompe péristaltique pour ensuite l'afficher sur l'interface sur le Raspberry Pi.

La calibration du YF-S402C sur le Raspberry Pi pour 1 litre, 98 tours complets de la turbine doit être faite.

```
1 from gpiozero import Button
2 from time import sleep
3
4 calibration = 98 # sert pour une calibration constante du YF-S402C
5                 # 98 = la calibration pour 1l pour le yf-s402c
6
7 compte = 0
8
9 def count_pulse():
10     global pulse_count
11     pulse_count += 1
12
13 flowsensor = Button(21, pull_up=True)
14 flowsensor.when_pressed = count_pulse
15
16 print("début du test...")
17
18 while True:
19     pulse_count = 0
20     sleep(1)
21     print(f"pluses/sec: {pulse_count} - débit estimé: {pulse_count / calibration * 60:.2f} l/min")
```

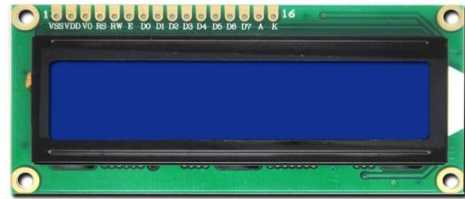
Shell x

```
début du test...
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
pluses/sec: 1 - débit estimé: 0.61 l/min
pluses/sec: 0 - débit estimé: 0.00 l/min
```

Figure 1 : Code de test pour le capteur de débit

### 3.3 Ecran LCD

Un écran LCD 16x02 avec I2C est un petit écran capable d'afficher du texte sur 16 colonnes et 2 lignes. Avec un module I2C, son utilisation est simplifiée. Grâce à ce module, l'écran n'a besoin que de 4 fils (VCC, GND, SDA et SCL) au lieu de beaucoup plus de câbles, ce qui rend le montage beaucoup plus simple. Il est contrôlé par un microcontrôleur qui



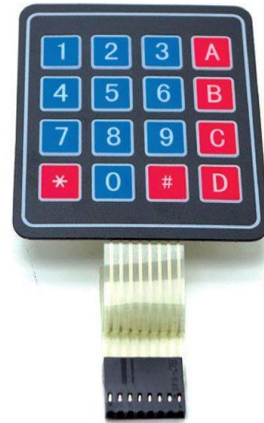
lui envoie les informations via le bus I2C. Cet écran est souvent utilisé pour afficher des données de capteurs, des messages ou des états de fonctionnement dans des projets électroniques, car il est pratique, économique et facile à câbler.

#### 3.3.1 Fonctionnement et choix dans le projet

Pour la pompe d'alimentation entérale, l'écran LCD sert d'affichage physique. Il affiche les données comme le débit et le ml/h. Pareil que sur l'interface, les informations dessus sont les mêmes et servent à montrer l'état de la pompe en cours.

## 3.4 Keypad

Un keypad 4x4 est un clavier matriciel composé de 16 touches disposées en 4 lignes et 4 colonnes. Il permet à l'utilisateur de saisir des chiffres, des lettres ou différentes commandes dans un système électronique. Chaque touche est reliée à une combinaison spécifique de ligne et de colonne. Lorsqu'une touche est pressée, un microcontrôleur détecte la ligne et la colonne correspondantes afin d'identifier la touche sélectionnée. Grâce à sa structure matricielle, le keypad nécessite moins de connexions qu'un clavier où chaque touche aurait son propre fil. Il est souvent utilisé dans les systèmes de contrôle d'accès par code, les alarmes, les serrures électroniques. Son principal avantage est qu'il permet une saisie simple et rapide des informations tout en restant peu coûteux et facile à intégrer dans un projet électronique.



### 3.4.1 Fonctionnement et choix dans le projet

Avec le même but que l'écran LCD 16x2, le keypad sert à insérer des informations de manière analogique. Le débit avec la touche « A » et/ou le volume avec la touche « B » peut être renseigné à l'aide du keypad. Ces informations sont alors enregistrées grâce à la touche « # » et ou supprimées avec la touche « \* ». Les données inscrites sur le keypad sont enregistrées et affichées dans l'interface également.

## 3.5 Bouton poussoir

Un bouton poussoir est un composant électronique qui permet d'ouvrir ou de fermer un circuit électrique lorsqu'il est pressé. Lorsque l'utilisateur appuie sur le bouton, le contact se fait et un signal électrique est envoyé au système. Lorsque le bouton est relâché, il revient automatiquement à sa position initiale grâce à un ressort interne et le contact est interrompu. Les boutons poussoirs sont très utilisés dans les projets électroniques pour déclencher une action, comme allumer un autre composant ou encore démarrer un programme ou envoyer une commande à un microcontrôleur. Ils sont appréciés pour leur simplicité, leur faible coût et leur facilité d'utilisation.



### 3.5.1 Fonctionnement et choix dans le projet

En plus des boutons dans l'interface, un bouton poussoir est ajouté pour démarrer ou arrêter la pompe. L'ajout d'un bouton physique assure l'état de la pompe si l'interface ne répond plus.

## 3.6 Buzzer piézoélectrique

Un buzzer piézoélectrique est un composant électronique capable de produire un son lorsqu'il reçoit un signal électrique. Il fonctionne grâce à un matériau piézoélectrique qui se déforme légèrement sous l'effet d'une tension électrique, créant ainsi des vibrations qui génèrent un son audible. Les buzzers piézoélectriques sont souvent utilisés pour les systèmes avec pour but d'émettre des signaux sonores ou des avertissements. Ils présentent l'avantage d'être peu coûteux, compacts, fiables et de consommer peu d'énergie, ce qui les rend très populaires dans de nombreux systèmes électroniques.



### 3.6.1 Fonctionnement et choix dans le projet

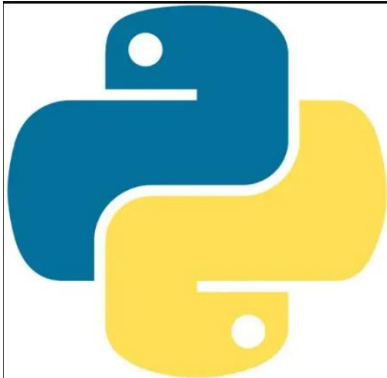
L'objectif du buzzer est, comme mentionné ci-dessus d'émettre un signal sonore pour avertir. Les pompes d'alimentations entérales en sont équipées pour prévenir le commencement, la fin ou un problème avec celle-ci. Son utilisation aura la même fonction pour connaitre l'état de la pompe sans aller sur l'interface pour le voir.

### **3.7 Alimentation 12v**

Une alimentation 12 V est utilisée pour fournir suffisamment d'énergie aux composants qui nécessitent une tension plus élevée que celle fournie par un microcontrôleur. Dans le cas d'un moteur pas-à-pas piloté par un A4988, une alimentation de 12 V permet d'obtenir un couple plus important et un fonctionnement plus stable, surtout lorsque le moteur tourne à une vitesse élevée. Le Raspberry Pi ou l'Arduino ne peuvent pas alimenter directement ce type de moteur, car ils fournissent généralement du 3,3 V ou du 5 V avec un courant limité. L'alimentation 12 V fournit donc l'énergie nécessaire au moteur, tandis que le contrôleur se contente d'envoyer les commandes de pilotage. Cela garantit de meilleures performances et évite de surcharger les circuits électroniques de commande.

## 4. Logiciel et technologie

### 4.1 Python



Python est un langage de programmation. Il est apprécié pour sa simplicité et est facile à comprendre, ce qui en fait un excellent choix pour les débutants comme pour les développeurs expérimentés. Python permet de créer de nombreux types d'applications, comme des sites web, des logiciels, des scripts d'automatisation, des programmes de traitement de données ou encore des projets d'intelligence artificielle. Dans les projets utilisant un Raspberry Pi, Python est souvent utilisé pour contrôler des capteurs, des moteurs et d'autres composants électroniques. Grâce à sa simplicité, sa polyvalence et sa grande communauté d'utilisateurs, Python est aujourd'hui l'un des langages de programmation les plus populaires au monde.

### 4.2 Flask



Flask est petit framework web Python. Il est léger et classé comme micro Framework et fournit des outils et fonctionnalités utiles à la création d'application Python. Flask a pour objectif de garder un noyau simple. Il n'a pas nativement un système d'authentification ni d'abstraction de base de données et outils de validation de formulaire mais pour autant de nombreuses extensions permettent d'ajouter facilement ces fonctionnalités.

Le choix de Flask a été pour ça facilité et gérer les différentes composantes de l'interfaces web de ma pompe. Grâce à cette technologie, la connexion avec le broker de MQTT, la base de données SQLite et les différentes pages HTML / CSS ont été centralisés avec le langage de programmation ; Python.

## 4.2.1 Fonctionnement dans le projet:

Avant d'arriver sur l'interface, l'utilisateur doit se connecter pour accéder à l'interface. Ces identifiants sont ensuite enregistrés dans la base de données. L'interface est filtrée par rapport à l'utilisateur.

Lorsque l'on appuie sur un bouton du site, la requête est alors envoyée et est reconnue grâce au GPIO. Si la pompe est connectée sur le GPIO 18, la requête est envoyée vers celui-ci.

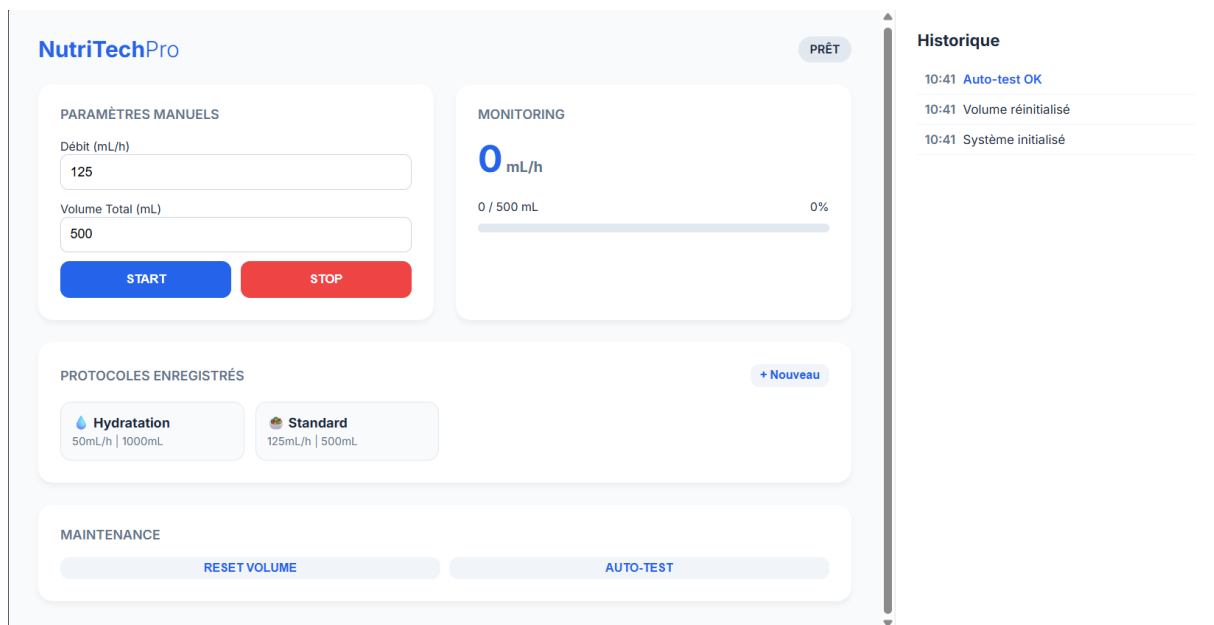


Figure 2: Interface web “NutriTechPro”

Ci-dessus se trouve l'interface de contrôle de la pompe d'alimentation NutriTechPro. Dans le premier cadran, paramètres manuels, on inscrit le débit auquel on veut que la pompe pousse le liquide et le volume à administrer. Juste à côté, on a le monitoring de la pompe pour voir l'état d'avancement.

Ensuite en dessous, il y a la possibilité d'enregistrer des pré-sélections pour pouvoir les lancer plus facilement. Avec le bouton Nouveau, on crée une nouvelle pré-sélection en donnant un titre et en ajoutant un débit et un volume.

Enfin, il y a la maintenance pour faire réinitialiser le volume. Sur le côté, il y a l'historique de toutes les actions qui sont enregistrées dans une table dans ma base de données.

## 4.3 MQTT



MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie qui se base sur un modèle de publication-abonnement. Il a été conçu pour faciliter la communication entre des appareils connectés pour des petits réseaux. Il est apprécié pour sa légèreté et sa capacité à s'adapter aux différentes contraintes de l'IoT.

Les appareils sont organisés en publisher (éditeur) et subscriber (abonné). Les éditeurs génèrent les données alors que les abonnés reçoivent ces données en s'abonnant au topic (sujet) qui les intéresse.

Le sujet est un canal de communication auquel un appareil peut s'abonner afin de recevoir le message qui a été publié dessus par l'éditeur. Grâce à cette architecture, les appareils n'ont pas besoin de maintenir une connexion constante avec d'autres appareils au sein du réseau. La complexité et la bande passante sont donc réduites.

```
juju@juju-raspberry:~/tfe-flask $ source nutritech_env/bin/activate
(nutritech_env) juju@juju-raspberry:~/tfe-flask $ pip install paho-mqtt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: paho-mqtt in ./nutritech_env/lib/python3.11/site-packages (2.1.0)
(nutritech_env) juju@juju-raspberry:~/tfe-flask $ |
```

Figure 3 : Installation de la bibliothèque paho-mqtt sur Raspberry-pi

### 4.3.1 Mosquitto

Eclipse Mosquitto est un broker qui fonctionne avec le protocole MQTT en open source.

Un publisher n'envoie jamais un message directement à un autre appareil.

Il publie son message dans un **topic** (un "sujet") le broker reçoit ce message et le transmet à tous les abonnés de ce topic. Les capteurs n'ont besoin de

savoir qui va lire les données et l'application n'a pas à savoir qui va envoyer.



### 4.3.2 Fonctionnement dans mon projet

Dans ce projet, Mosquitto est installé sur le Raspberry Pi et sert de **broker** MQTT. Le Raspberry Pi envoie les informations de la pompe sur plusieurs **topics** MQTT : **pompe/debit** pour le débit mesuré par le capteur, **pompe/volume** pour le volume administré, ainsi que **pompe/start** et **pompe/end** pour indiquer respectivement le démarrage et l'arrêt de la pompe. L'application **Flask** s'abonne à ces différents topics afin de recevoir les données en temps réel. Ces informations sont ensuite utilisées pour mettre à jour et afficher les données de fonctionnement de la pompe sur le tableau de bord **NutritechPro**.

```
MQTT_BROKER = "localhost" # Remplace
MQTT_PORT = 1883
TOPIC_DEBIT = "pompe/debit"
TOPIC_VOLUME = "pompe/volume"
TOPIC_START = "pompe/start"
TOPIC_END = "pompe/end"
```

Figure 4 : Topics

## 4.4 SQLite



SQLite est un système de gestion de base de données relationnelle léger. Il est autosuffisant et ne dépend pas d'un système client-serveur typique.

Différent des bases de données de MySQL ou MariaDB, SQLite ne dépend pas d'un serveur pour fonctionner. Il est directement intégré dans le programme. SQLite écrit ces données SQL dans un fichier juste à côté du programme.

Pour un Raspberry Pi, SQLite est une bonne option pour plusieurs raisons.

Tout d'abord, il a une surcharge assez peu élevée ainsi qu'un système autonome. Aucune dépendance n'est nécessaire pour assurer son fonctionnement. Son utilisation est très simple et prête à l'emploi juste après son installation.

Ensuite, étant donné qu'il n'y a pas de processus serveur séparé, SQLite ne consomme ni la RAM ni le processeur du Raspberry Pi lors de son utilisation.

### 4.4.1 Choix

Le choix de cette solution était pour plusieurs raisons. Créer une base de données pour les utilisateurs et les actions de la pompe.

Premièrement, sa rapidité d'installation et sa mise à l'emploi sont très avantageuses sur un Raspberry Pi. Faire une liaison avec les composants et l'interface devient plutôt simple.

Ensuite, ayant déjà utilisé SQLite sur un projet l'année précédente sur un Raspberry Pi, j'ai trouvé ça judicieux d'utiliser encore le programme.

En bref, grâce à sa mise à l'emploi simple, sa rapidité ainsi qu'une utilisation ultérieure, j'ai préféré choisir cette solution plutôt qu'une autre.

## 4.4.2 Fonctionnement dans le projet :

Comme cité plus tôt, j'ai créé une base de données avec plusieurs tables.

La première table est destinée à l'inscription et à la connexion des utilisateurs.

```
##### Mot de passe oublié #####
@app.route('/password_reset', methods=['GET', 'POST'])
def reset():
    if request.method == 'POST':
        email = request.form['email']
        new_password = request.form['new_password']

        conn = get_db()
        try:
            user = conn.execute('SELECT id FROM users WHERE email = ?', (email,)).fetchone()

            if not user:
                return render_template('reset.html', erreur="Aucun compte avec cet email")

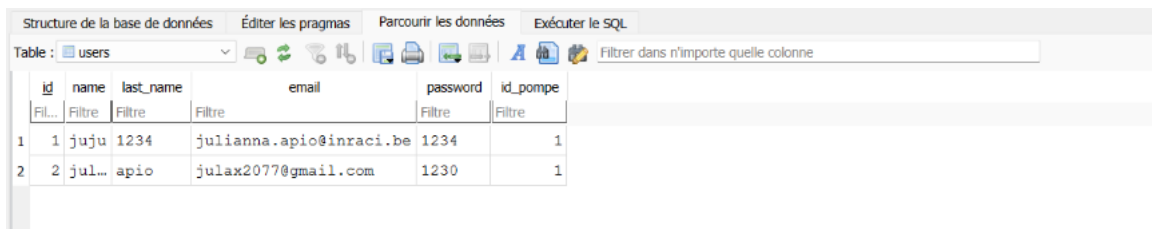
            conn.execute(
                (
                    'UPDATE users SET password = ? WHERE email = ?',
                    (new_password, email)
                )
            )
            conn.commit()
        finally:
            conn.close()

        return redirect(url_for('login'))

    return render_template('password_reset.html')
```

Figure 5 : Bout de code pour modifier le mdp

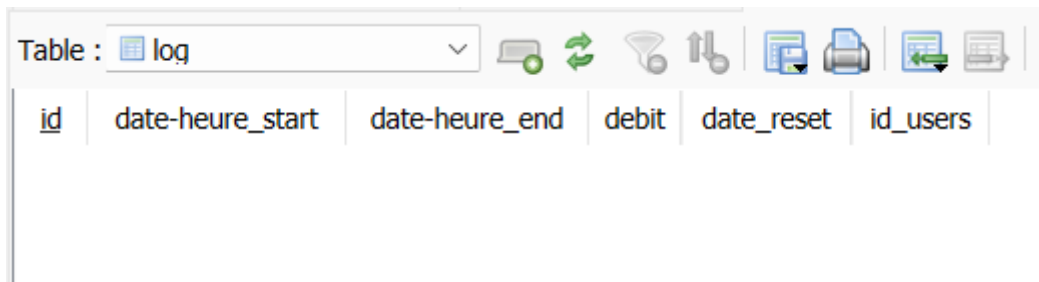
Cette fonction permet de changer de mot de passe lorsqu'on l'oublie. L'utilisateur y insère son email et un nouveau mot de passe dans les entrées textes sur une page HTML et ensuite ils sont récupérés. Le mot de passe est directement modifié dans la table user dans la colonne password.



id	name	last_name	email	password	id_pompe
1	juju	1234	julianna.apio@inraci.be	1234	1
2	jul...	apio	julax2077@gmail.com	1230	1

Figure 6 : Table SQLite des utilisateurs

La deuxième table, elle sert de journal d'événements des actions réalisées par pompe. La date de l'heure de lancement et d'arrêt sont enregistrées dans cette table et seront ensuite affichées dans l'historique sur l'interface.

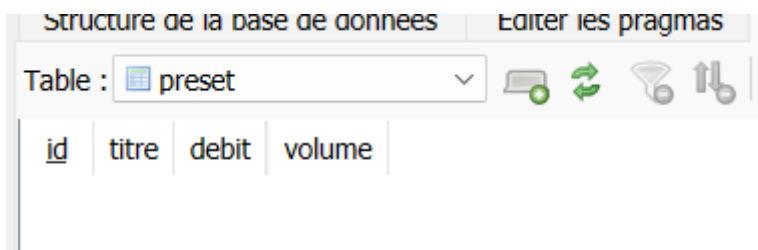


The screenshot shows a SQLite table named 'log'. The table has six columns: 'id', 'date-heure\_start', 'date-heure\_end', 'debit', 'date\_reset', and 'id\_users'. The 'id' column is underlined, indicating it is the primary key. The interface includes a dropdown menu for the table name and several icons for database operations like refresh, refresh with error, refresh with warning, refresh with success, save, print, and export.

<u>id</u>	date-heure_start	date-heure_end	debit	date_reset	id_users
-----------	------------------	----------------	-------	------------	----------

Figure 7 : Table SQLite des logs

La dernière table a pour but d'enregistrer les présélections créées par l'utilisateur, qui sont enregistrées dans cette table. Le titre, le débit ainsi que le volume sont stockés pour être ensuite utilisés avec la pompe d'alimentation.



The screenshot shows a SQLite table named 'preset'. The table has four columns: 'id', 'titre', 'debit', and 'volume'. The 'id' column is underlined, indicating it is the primary key. The interface includes a dropdown menu for the table name and several icons for database operations like refresh, refresh with error, refresh with warning, refresh with success, save, print, and export.

<u>id</u>	titre	debit	volume
-----------	-------	-------	--------

Figure 8 : Table SQLite des présélection

## 4.5 Tailscale



Tailscale est une solution de VPN moderne et s'appuie sur le protocole WireGuard. WireGuard est une alternative à d'autres protocoles VPN comme OpenVPN. La particularité de Tailscale est qu'il soit open-source, performant. Il est également flexible et s'adapte à divers environnements sur une multitude d'appareils, que ce soit un PC, un NAS ou même un serveur. L'offre gratuite est limitée à 3 utilisateurs et peut accueillir jusqu'à 100 appareils.

Tailscale fonctionne selon un modèle peer-to-peer (pair à pair). Il fonctionne en créant un réseau maillé où chaque appareil peut communiquer directement avec d'autres appareils. Peu importe où un appareil se trouve, une communication peut être faite sans avoir besoin d'ouvrir un port et de l'exposer. Les nœuds connectés à du VPN sont appelés Tailnet.

Comparé aux VPN classiques, Tailscale s'appuie sur un système d'authentification d'identité cloud externe avec, par exemple, Google, Microsoft, ou encore GitHub et Apple.

### 4.5.1 Fonctionnement et choix dans mon projet

Tailscale a été installé et utilisé dans ce projet pour régler un problème : créer un accès sur un réseau distant de façon sécurisée.

L'objectif du projet est de permettre une connexion à distance sur le Raspberry Pi afin d'accéder à l'interface web de pompe et de prendre le contrôle. Avec Tailscale, cette solution est possible. IL va créer un tunnel chiffré de bout en bout entre 2 appareils pour leur faire croire qu'ils sont dans le même réseau local sans avoir besoin d'ouvrir des ports sur son routeur.

```
juju@juju-raspberry:~$ tailscale status
100.79.145.112 juju-raspberry julax2077@ linux -
100.101.11.125 juju-telephone julax2077@ android -
100.126.205.76 pc-julianna julax2077@ windows active; direct 192.168.1.29:41641, tx 12084 rx 11580
100.110.128.119 pc-maison julax2077@ windows offline, last seen 31d ago
juju@juju-raspberry:~$ |
```

Figure 9 : Statut de Tailscale sur le Raspberry pi

## 4.6 Nginx



Nginx (prononcé “engine-ex”) est un serveur web open-source créé en 2004. Il sert à afficher des sites web, mais aussi à jouer d’autres rôles importants comme **reverse proxy**, **cache HTTP** ou encore **équilibreur de charge (load balancer)**.

Il est très utilisé par de grandes entreprises car il peut gérer beaucoup de visiteurs en même temps tout en utilisant peu de mémoire. Pour cela, il fonctionne avec un système asynchrone et événementiel, ce qui veut dire qu’il traite les requêtes sans bloquer les autres. Un processus principal gère plusieurs processus secondaires qui s’occupent des requêtes en parallèle.

En résumé : Nginx est un logiciel très performant qui permet de servir des sites web rapidement et efficacement, même avec beaucoup de trafic.

### 4.6.1 Fonctionnement dans mon projet

Nginx est solution plus rapide et simple d’installation que apache2 pour utiliser un Raspberry pi comme serveur web.

Statu de Nginx sur le Raspberry

```
juju@juju-raspberry:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Sat 2026-05-02 01:25:24 CEST; 4min 50s ago
     Docs: man:nginx(8)
  Process: 4204 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 4205 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 4233 (nginx)
   Tasks: 5 (limit: 8743)
    CPU: 61ms
  CGroup: /system.slice/nginx.service
          └─4233 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
            └─4235 "nginx: worker process"
              └─4236 "nginx: worker process"
                └─4237 "nginx: worker process"
                  └─4238 "nginx: worker process"

mai 02 01:25:24 juju-raspberry systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
mai 02 01:25:24 juju-raspberry systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
juju@juju-raspberry:~$
```

Figure 10 : \_Statu de Nginx sur le Raspberry pi

## 4.7 Cloudflare



**CLOUDFLARE**

Cloudflare est une entreprise qui fournit des services permettant d'améliorer la sécurité et les performances des sites web et des applications. Grâce à un grand réseau de centres de données répartis dans de nombreuses villes à travers le monde, elle permet de rapprocher le contenu des utilisateurs, ce qui réduit le temps de chargement des pages web. Cloudflare protège également les sites contre différentes menaces informatiques, comme les attaques DDoS et les robots malveillants, afin de garantir leur bon fonctionnement. En plus de ces services, l'entreprise propose un service DNS gratuit appelé **1.1.1.1**, qui aide à protéger la vie privée des utilisateurs en limitant la collecte et l'utilisation de leurs données de navigation.

### 4.6.1 Cloudflare Tunnel

**Cloudflare Tunnel** est un service qui permet de rendre un serveur local accessible sur Internet de façon **sécurisée**, sans ouvrir de ports sur le routeur. Cela est plus sûr, car l'adresse IP du serveur n'est jamais directement visible sur Internet.

Pour fonctionner, il utilise un programme appelé **cloudflared**, installé sur le serveur. Ce programme crée une connexion sortante vers Cloudflare. Ensuite, quand un utilisateur veut accéder au service, sa requête passe d'abord par Cloudflare, puis est envoyée au serveur local via ce tunnel sécurisé.

En clair, Cloudflare Tunnel sert à **accéder à un serveur privé depuis Internet sans exposer le réseau** et en améliorant la sécurité.

## 4.6.2 Fonctionnement et choix dans le projet

Pour ne pas avoir à acheter un VPS, l'utilisation de Cloudflare est une solution gratuite et simple d'installation. L'accès est possible grâce à une URL publique.

Quand un utilisateur veut accéder au Dashboard sans être sur le réseau local, Cloudflare intervient et redirige vers le Raspberry Pi grâce à un tunnel sécurisé. Nginx, lui, reçoit la requête puis la transmet à l'application Flask qui génère et renvoie la page web.

En utilisant cette configuration, un accès sur le Dashboard de la pompe d'alimentation est possible sur n'importe quel réseau distant de manière sécurisée et sans une configuration réseau complexe.

```
juju@juju-raspberry:~ $ cloudflared tunnel route dns nutritech home.nutritec
h.be --overwrite-dns
This command expects the format "cloudflared tunnel route dns <tunnel name/i
d> <hostname>"
See 'cloudflared route dns --help'.
juju@juju-raspberry:~ $ cloudflared tunnel route dns nutritech home.julianna
apiomakani.be
2026-05-01T23:07:22Z INF Added CNAME home.juliannaapiomakani.be which will r
oute to this tunnel tunnelID=1b32880c-a576-497d-b0e1-a0ae945af844
juju@juju-raspberry:~ $
```

Figure 11 : Création du tunnel appelé **nutritech** sur le Raspberry pi

Serveurs DNS

Configuration DNS personnalisée ?  
 Non  Oui

Serveurs DNS 1: donovan.ns.cloudflare.co

Serveurs DNS 2: jen.ns.cloudflare.com

Serveurs DNS 3:

Serveurs DNS 4:

Ipv4 1:

Ipv4 2:

Ipv4 3:

Ipv4 4:

Valider

Domaine(s) & Hébergement(s)							Serveurs	Stockage en ligne	Autres services	
15	Enregistrements par page						Rechercher <input type="text"/>			
<input type="checkbox"/>	Domaine	Registrar	Infos	Formule	Expiration	Action				
<input type="checkbox"/>	juliannaapiomakani.be	oui	Dns Lws	domaine	14-10-2026	Gérer Renouveler				
<input type="checkbox"/>	nutritech.be	oui	Dns Cloudflare	domaine	30-04-2027	Gérer Renouveler				

Renouveler la sélection

< Précédent 1 Suivant >

Figure 12 & 13: Changement des DNS de « nutritech.be » vers ceux de Cloudflare

#### 4.6.3 Problèmes et solutions rencontrés :

##### Problème :

“cloudflared tunnel route dns nutritech [nutritech.be](https://nutritech.be)”

Après avoir changé les serveurs DNS pour qu’il soit géré par cloudflared et non par lws, je n’avais pas ajouté le domaine dans Cloudflare et en remettant la commande ci-dessus, j’avais une erreur.

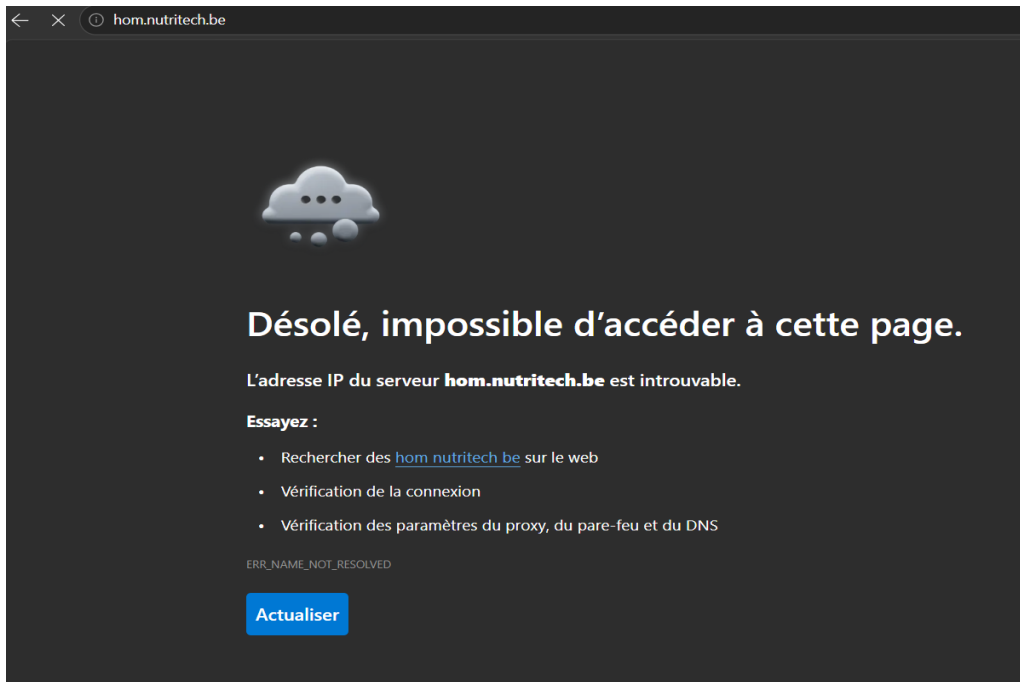


Figure 14: Navigation vers nutritech.be

Solution : ajouter le nom de domaine [nutritech.be](https://nutritech.be) à Cloudflare pour que le tunnel puisse rediriger vers « [nutritech.be](https://nutritech.be) » comme ça a été défini dans le fichier de configuration de cloudflared.

```
tunnel: nutritech
credentials-file: /home/juju/.cloudflared/1b32880c-a576-497d-b0e1-a0ae945af
ingress:
- hostname: nutritech.be
  service: http://localhost:80
- service: http_status:404
```

Figure 15 : Configuration du tunnel Cloudflare sur le Raspberry Pi

## 4 Architecture du système

### 5.1 Topologie Cloudflare

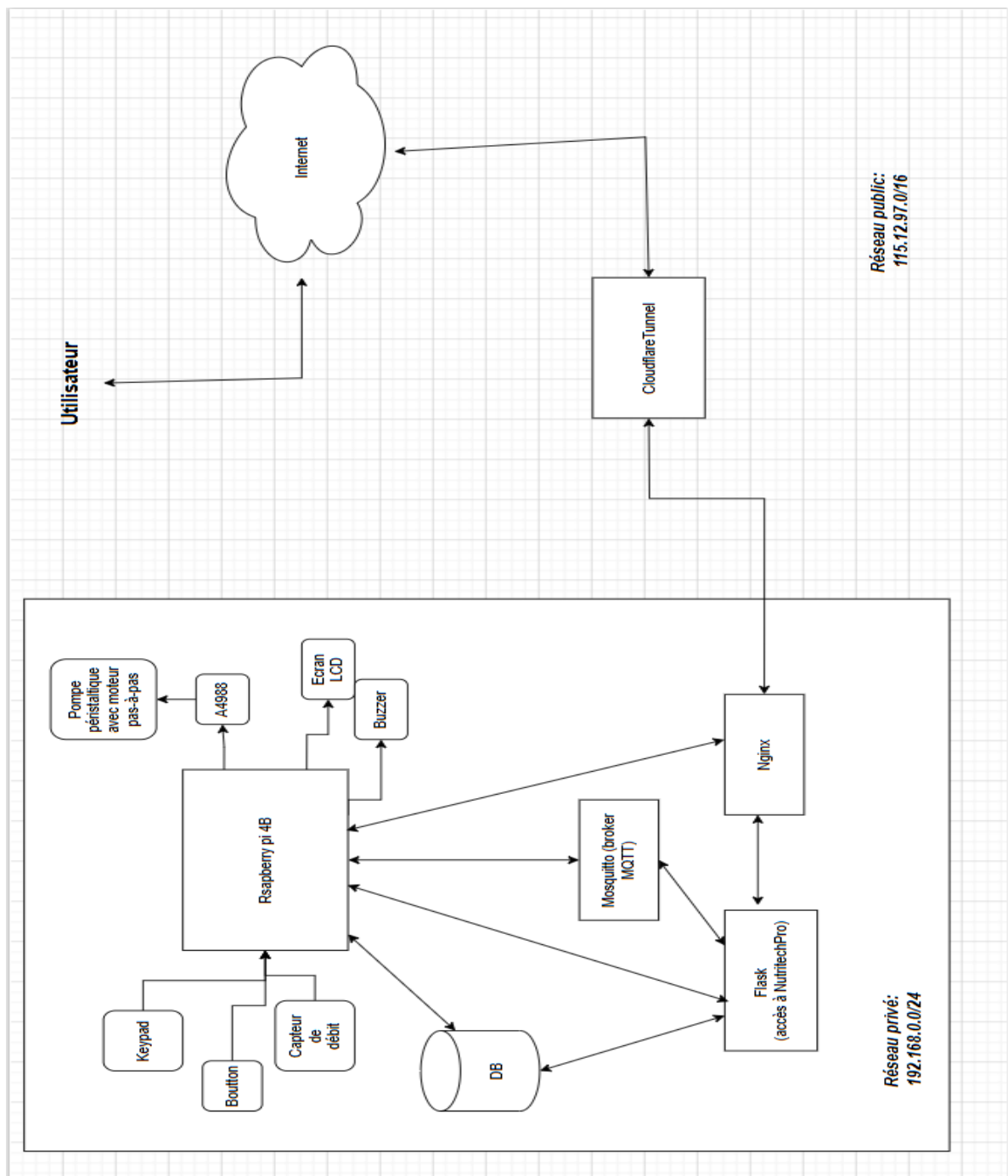


Figure 16: Topologie du système NutritechPro — accès via Cloudflare Tunnel

La figure X illustre l'architecture réseau du système quand un utilisateur se connecte au Dashboard (NutriTechPro). Le Raspberry Pi héberge l'application Flask accessible depuis Internet via Cloudflare Tunnel, sans exposition directe de l'adresse IP du serveur.

## 5.2 Schéma électronique

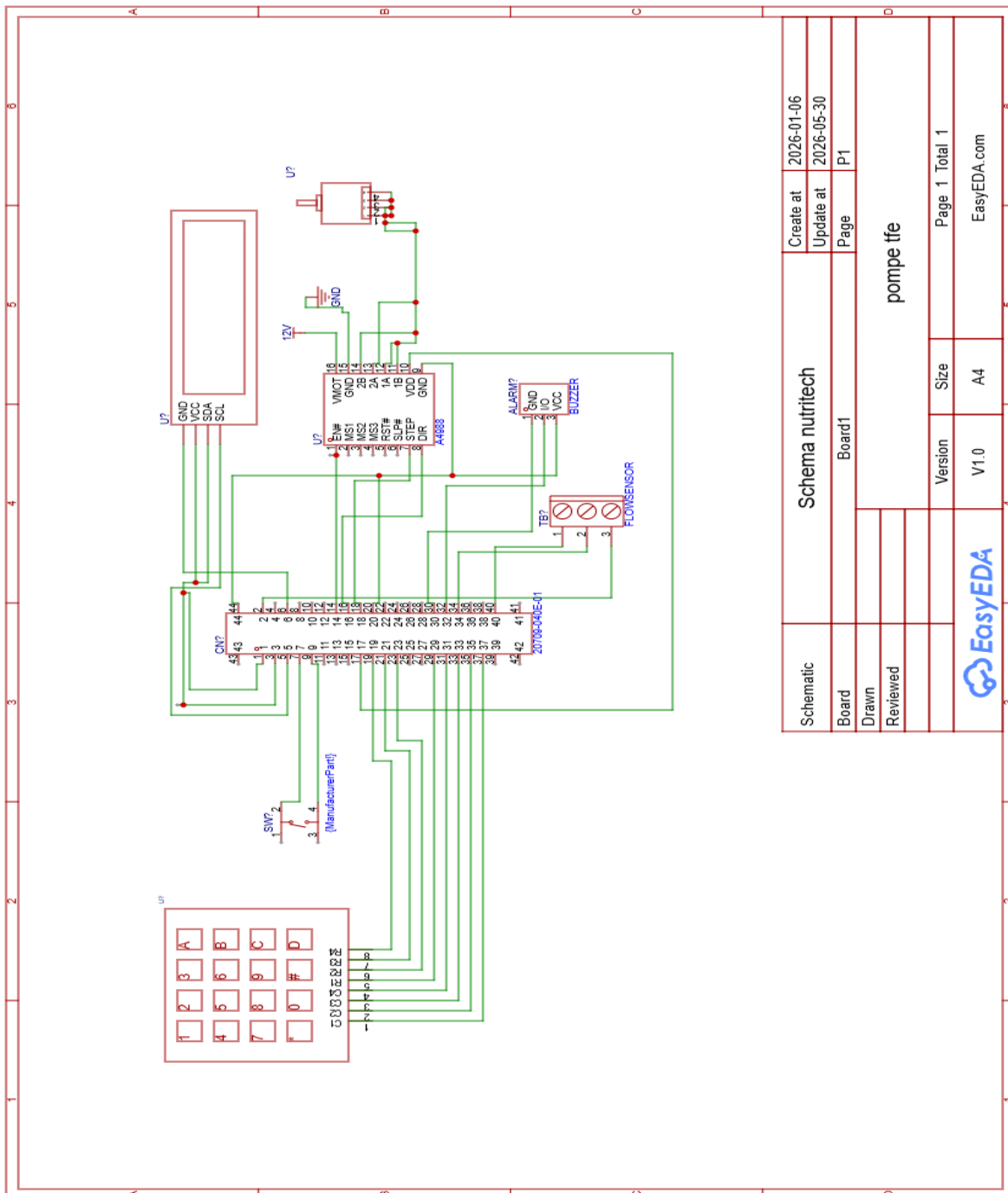


Figure 17 : Schéma électronique de NutriTech

Le schéma ci-dessus représente les connexions électroniques entre le Raspberry Pi 4B et les différents composants du système. Le connecteur CN1 représente les broches GPIO du Raspberry Pi. L'alimentation 12V externe alimente le driver A4988 pour le moteur pas-à-pas.

### 5.3 Schéma mécanique

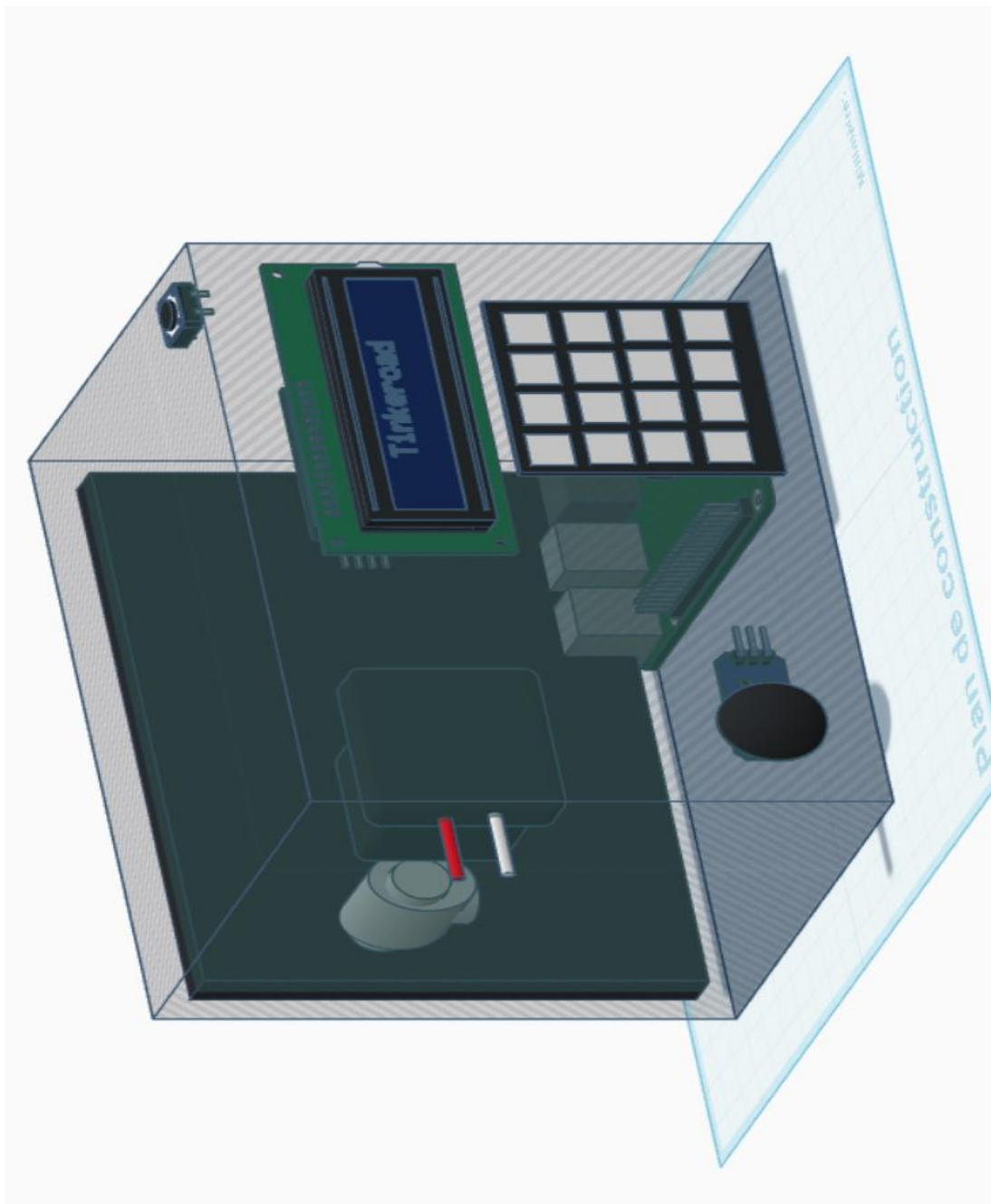


Figure 18 : Vue 3D du boîtier du système Nutritech (Tinkercad)

La figure X présente une vue d'ensemble du boîtier conçu sur Tinkercad. Elle illustre le placement des différents composants à l'intérieur du boîtier avant impression 3D.

## 5. Conclusion

En résumé, le travail réalisé a été la configuration des divers composants pour créer la pompe d'alimentation, la réalisation du dashboard avec Flask afin de permettre le contrôle de celle-ci et l'installation de différents logiciels pour son bon fonctionnement.

Au moment où ce rapport a été rédigé, le projet n'est pas totalement opérationnel. Plusieurs drivers pour la pompe avec moteur pas-à-pas ont été grillés ou sont défectueux. Des problèmes de santé ainsi que de mauvais avis médicaux m'ont ralenti pendant plusieurs semaines.

Malgré tous ces problèmes, NutriTech est un projet que j'apprécie beaucoup. Pouvoir aider mon frère et peut-être d'autres personnes me tient à cœur. Ma mère a parlé de mon projet à des personnes qui s'occupent de mon frère et qui connaissent des gens qui travaillent pour donner des pompes d'alimentation. Ils m'ont dit que je pourrais proposer cette idée et qu'elle pourrait aider de nombreuses personnes pour leur alimentation quotidienne.

## 6. Bibliographie

OBJETCONNECTE, « Tout savoir sur le protocole MQTT et ses applications », sur <https://www.objetconnecte.com/tout-savoir-sur-le-protocole-mqtt-et-ses-applications/>, mis à jour le 19 août 2025, consulté le 22 mai 2026

RASPBERRY-PI FRANCE, “Installing Tailscale to the Raspberry Pi - Pi My Life Up”, sur <https://pimylifeup.com/raspberry-pi-tailscale/>, 28 février 2026, consulté le 24 avril 2026

Emmet, “Qu’est-ce qu’un Raspberry Pi ? - Raspberry Pi France”, sur <https://www.raspberrypi-france.fr/guide/quest-ce-qu'un-raspberry-pi/>, consulté le 29 avril 2026

WIKIPEDIA, “Flask (framework)”, sur ["wikipedia.org/wiki/Flask\\_\(framework\)"](https://fr.wikipedia.org/wiki/Flask_(framework)), 22 avril 2026, consulté le 23 avril 2026

Dhairya Parikh, “Utiliser Flask sur le Raspberry Pi : un guide complet “, sur <https://raspberrytips.com/using-flask-on-raspberry-pi/>, 13 novembre 2024, consulté le 22 avril 2026

Emmet, “Installer un tunnel Cloudflare sur le Raspberry Pi “, sur <https://pimylifeup.com/raspberry-pi-cloudflare-tunnel/>, 7 novembre 2025, consulté le 1 mai 2026

SQLITE, “Introduction à SQL et SQLite”, sur <https://sqlite.fr/>, consulté le 19 mai 2026

Emmet, “Mettre en place une base de données SQLite sur un Raspberry Pi”, sur <https://pimylifeup.com/raspberry-pi-sqlite/>, 19 janvier 2022, consulté le 22 mai 2026

CLOUDFLARE, “Qu'est-ce que Cloudflare ?”, sur <https://www.cloudflare.com/fr-fr/learning/what-is-cloudflare/>, consulté le 22 mai 2026

Krodelabestiole, “samba “, sur <https://doc.ubuntu-fr.org/samba>, 28 décembre 2025, consulté le 30 mai 2026

NEWBIELY, “Raspberry pi –LCD”, sur <https://newbiely.fr/tutorials/raspberry-pi/raspberry-pi-lcd/>, consulté le 10 décembre 2026

NEWBIELY, “Raspberry pi –Keypad”, sur <https://newbiely.com/tutorials/raspberry-pi/raspberry-pi-keypad/>, consulté le 10 décembre 2026

NEWBIELY, "Raspberry pi – Piezo Buzzer", sur <https://newbiely.com/tutorials/raspberry-pi/raspberry-pi-piezo-buzzer/>, consulté le 10 décembre 2026

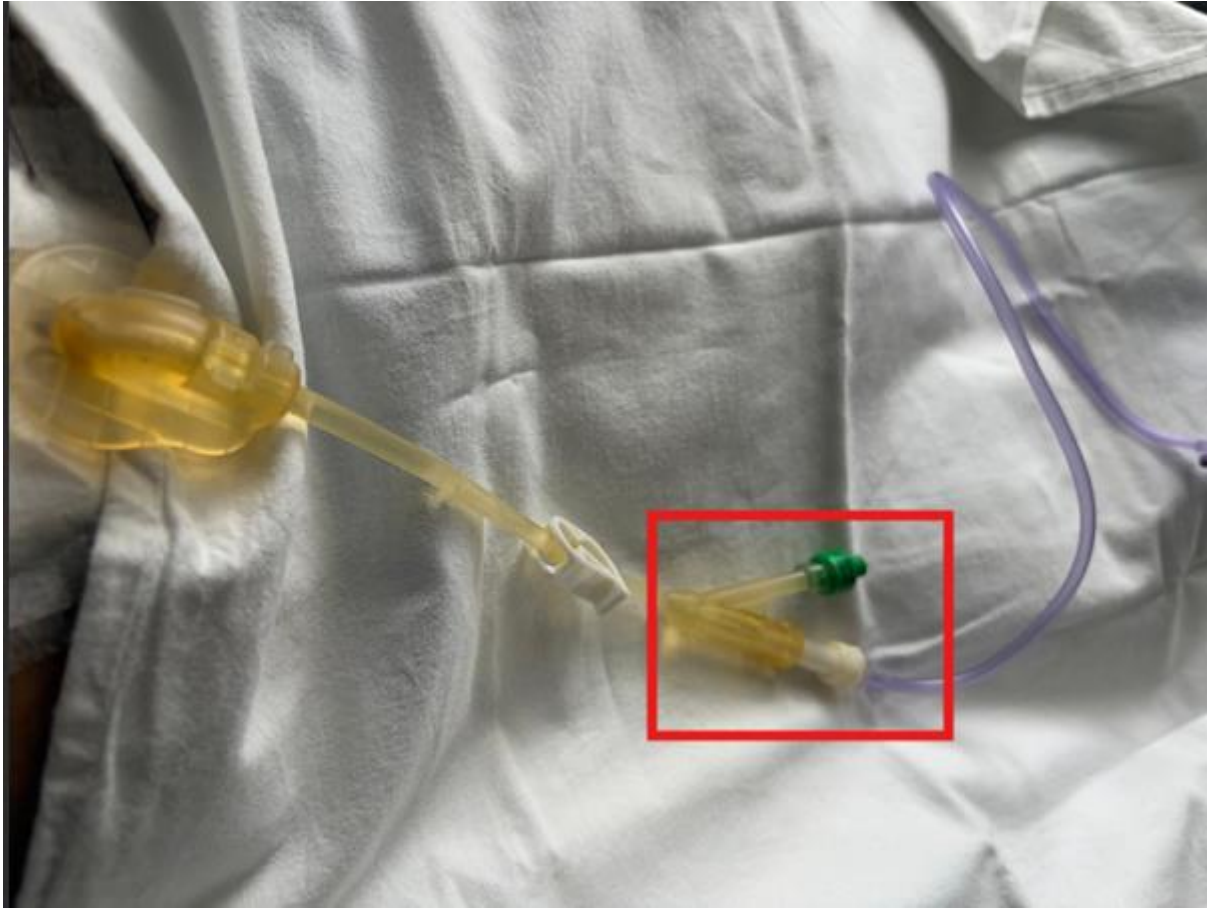
## 7. Annexe



Figure 19 : Pompe d'alimentation vendu sur le marché et utilisée par mon frère.



Figure 19 : Poche avec le liquide qui va être pompé par la pompe d'alimentation ci-dessus.



*Figure 20 : Sonde gastrique et enbout de tube de la pompe par laquelle passe le liquide*

```

juju@juju-raspberry:~/tfe-flask $ python3 -m venv nutritech_env
juju@juju-raspberry:~/tfe-flask $ ls
nutritech_env
juju@juju-raspberry:~/tfe-flask $ source nutritech_env/bin/activate
(nutritech_env) juju@juju-raspberry:~/tfe-flask $ pip3 install Flask
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting Flask
  Downloading https://www.piwheels.org/simple/flask/flask-3.1.3-py3-none-any.whl (103 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 103.4/103.4 kB 1.3 MB/s eta 0:00:00
Collecting blinker>=1.9.0
  Downloading https://www.piwheels.org/simple/blinker/blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting click>=8.1.3
  Downloading https://www.piwheels.org/simple/click/click-8.3.3-py3-none-any.whl (110 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.5/110.5 kB 4.0 MB/s eta 0:00:00
Collecting itsdangerous>=2.2.0
  Downloading https://www.piwheels.org/simple/itsdangerous/itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting jinja2>=3.1.2
  Downloading https://www.piwheels.org/simple/jinja2/jinja2-3.1.6-py3-none-any.whl (134 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.9/134.9 kB 4.7 MB/s eta 0:00:00
Collecting markupsafe>=2.1.1
  Downloading markupsafe-3.0.3-cp311-cp311-manylinux2014_aarch64.manlinux_2_17_aarch64.manlinux_2_28_aarch64.whl (24 kB)
Collecting werkzeug>=3.1.0
  Downloading https://www.piwheels.org/simple/werkzeug/werkzeug-3.1.8-py3-none-any.whl (226 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 226.5/226.5 kB 1.8 MB/s eta 0:00:00
Installing collected packages: markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, Flask
Successfully installed Flask-3.1.3 blinker-1.9.0 click-8.3.3 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 werkzeug-3.1.8
(nutritech_env) juju@juju-raspberry:~/tfe-flask $ |

```

Figure 21 : Création d'un environnement pour l'installation de Flask sur Raspberry Pi

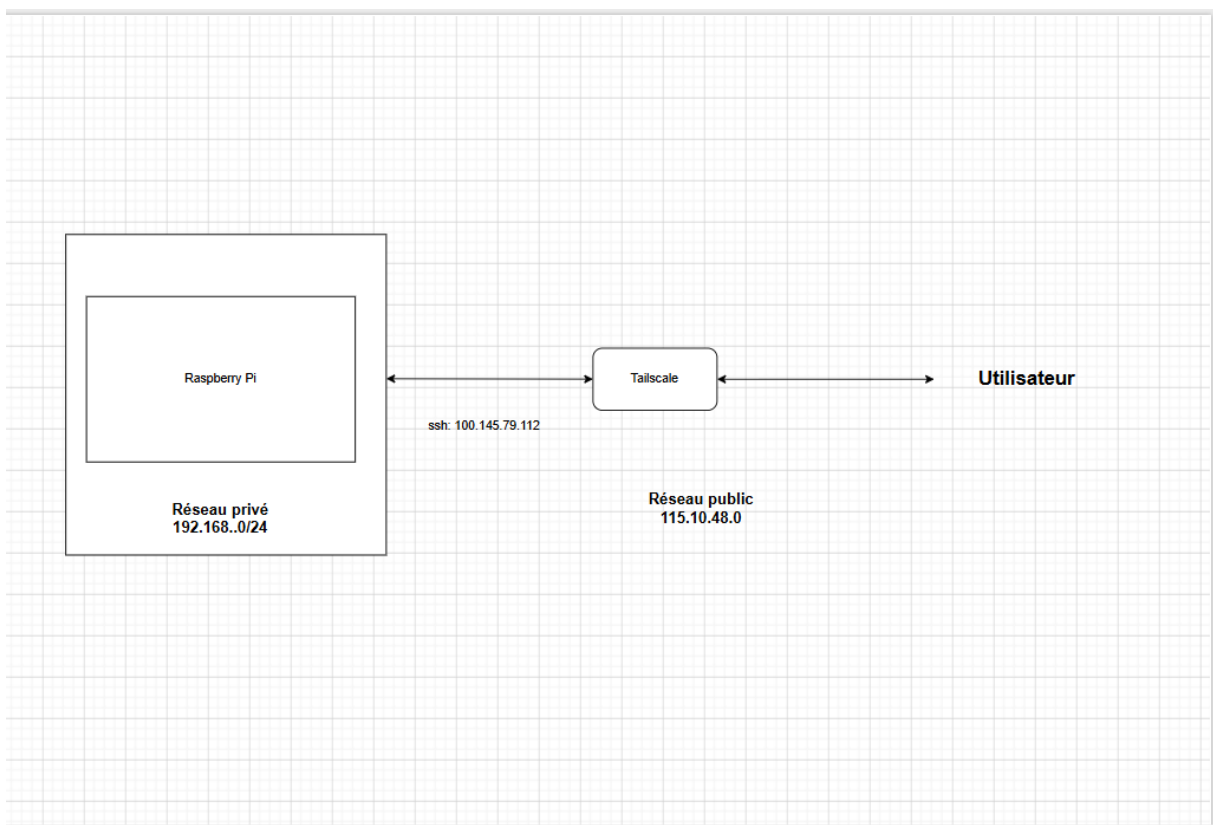


Figure 22 : Topologie de Tailscale

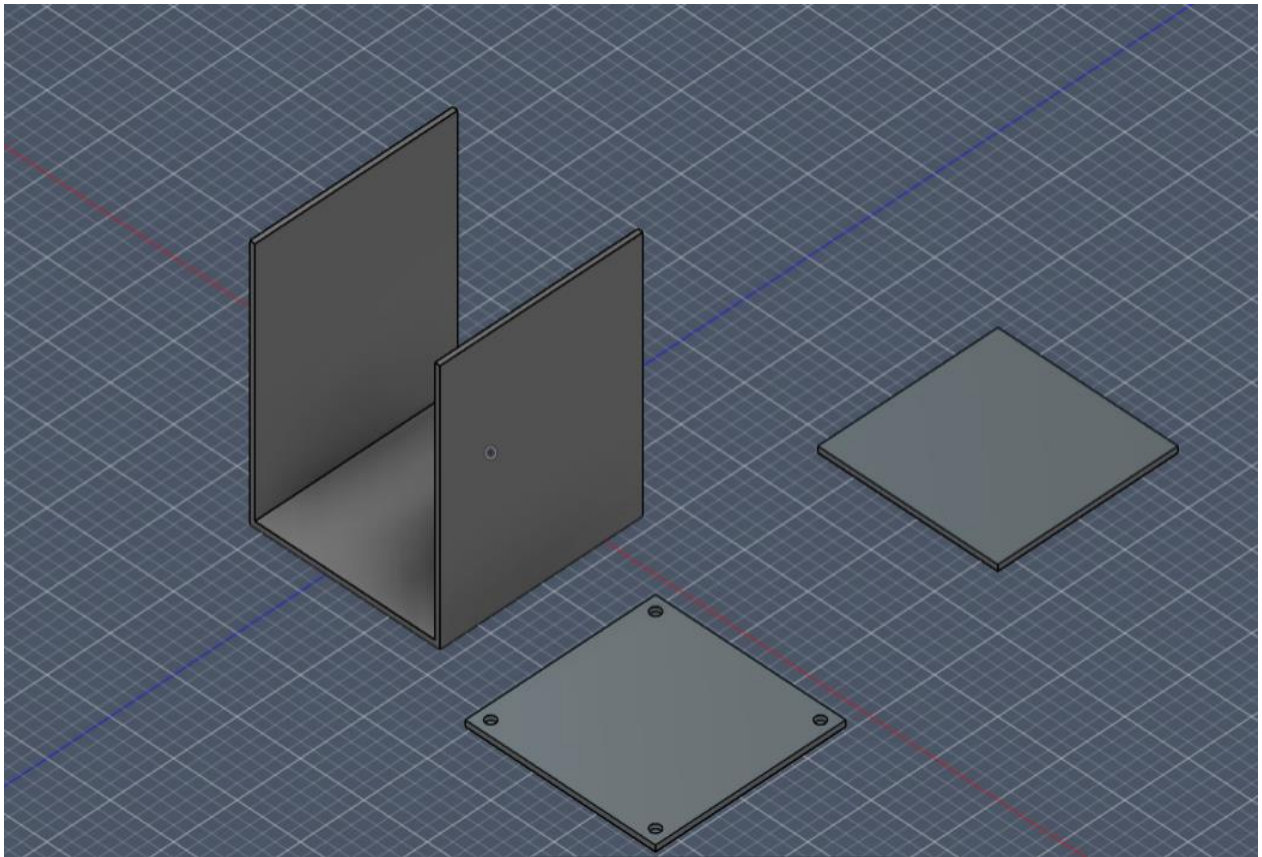


Figure 23 : Conception 3D sur fusion 360 à imprimer (15x15x15)

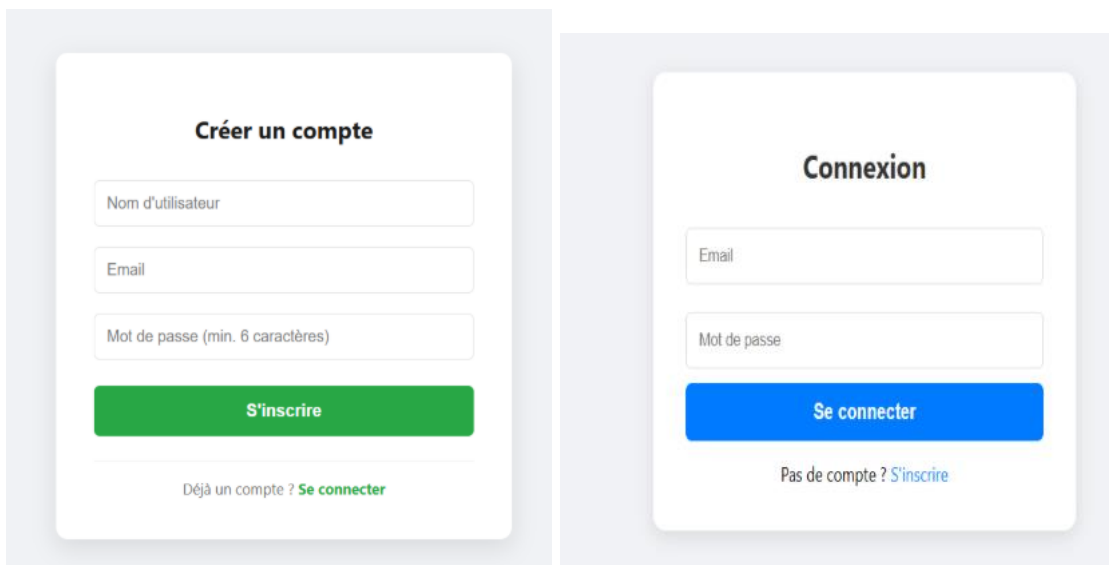
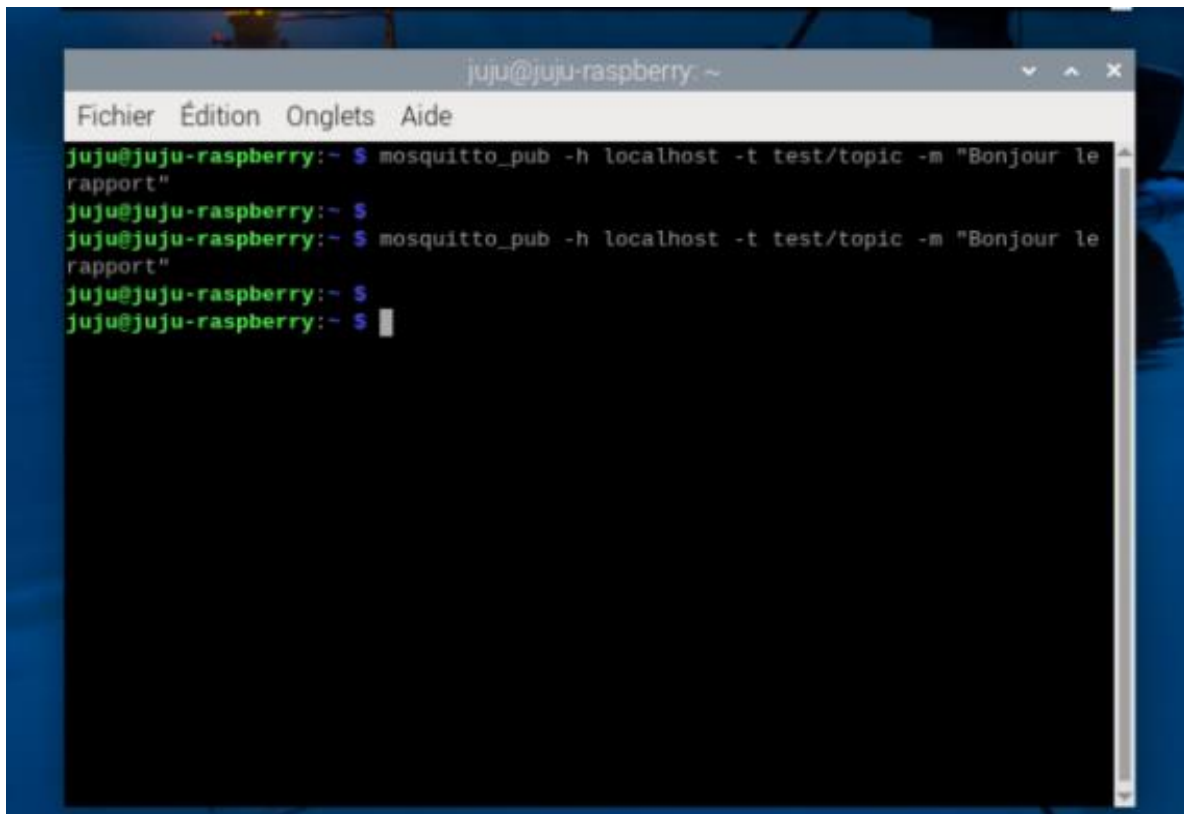
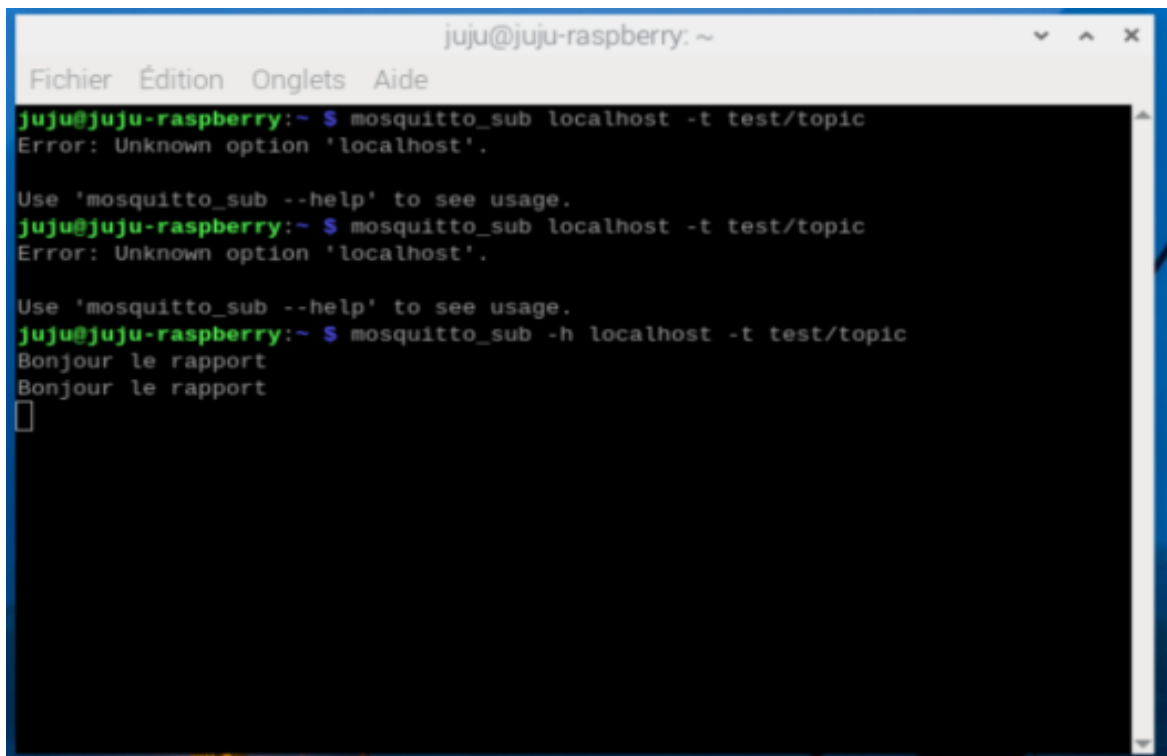


Figure 24 & 25 : Page d'inscription et de connexion au Dashboard

A terminal window titled 'juju@juju-raspberry: ~' with a menu bar containing 'Fichier', 'Édition', 'Onglets', and 'Aide'. The terminal shows four lines of command execution:

```
juju@juju-raspberry:~ $ mosquitto_pub -h localhost -t test/topic -m "Bonjour le rapport"
juju@juju-raspberry:~ $
juju@juju-raspberry:~ $ mosquitto_pub -h localhost -t test/topic -m "Bonjour le rapport"
juju@juju-raspberry:~ $
juju@juju-raspberry:~ $
```

Comme le montre cette image, je teste MQTT. La commande **mosquitto\_pub** permet de publier un topic sur le broker. Le topic test/topic est celui que j'ai créé avec comme contenu « Bonjour le rapport en local ».

A terminal window titled 'juju@juju-raspberry: ~' with a menu bar containing 'Fichier', 'Édition', 'Onglets', and 'Aide'. The terminal shows three commands and their outputs. The first two commands are 'mosquitto\_sub localhost -t test/topic', both resulting in 'Error: Unknown option 'localhost''. The third command is 'mosquitto\_sub -h localhost -t test/topic', which results in 'Use 'mosquitto\_sub --help' to see usage.' followed by 'Bonjour le rapport' appearing twice on separate lines, and a cursor on the third line.

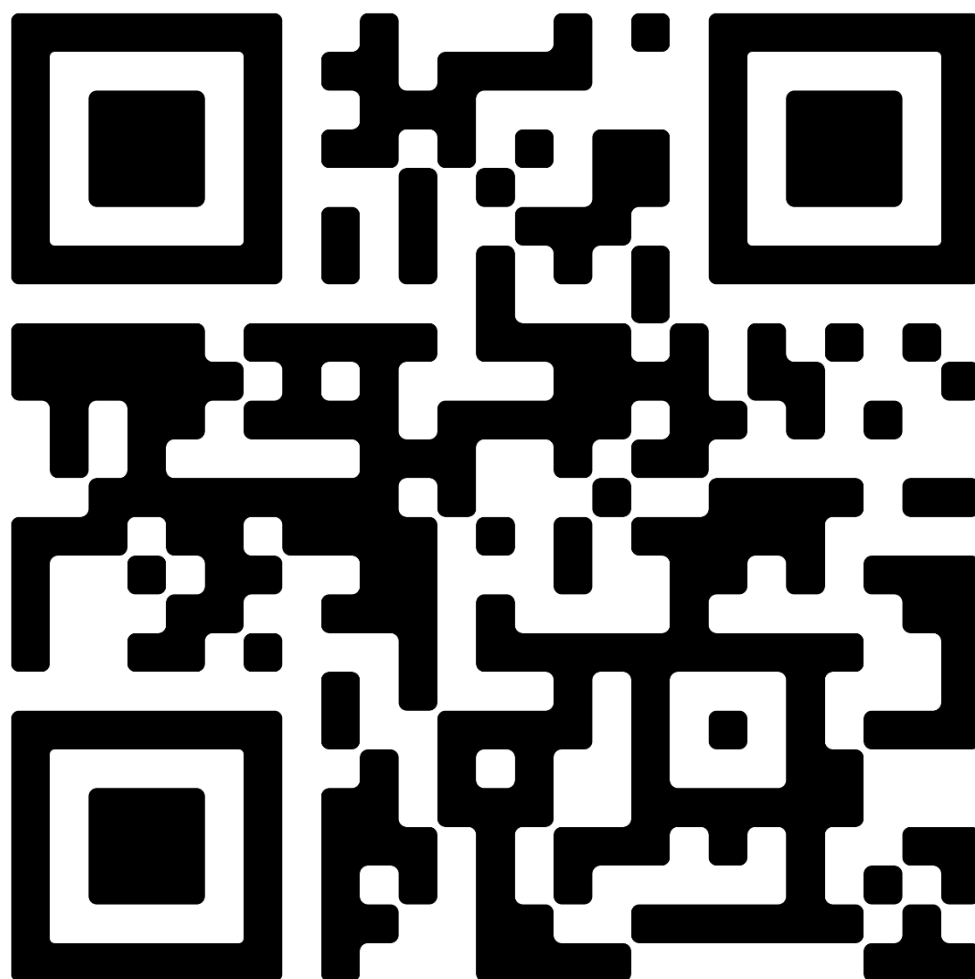
```
juju@juju-raspberry:~ $ mosquitto_sub localhost -t test/topic
Error: Unknown option 'localhost'.

Use 'mosquitto_sub --help' to see usage.
juju@juju-raspberry:~ $ mosquitto_sub localhost -t test/topic
Error: Unknown option 'localhost'.

Use 'mosquitto_sub --help' to see usage.
juju@juju-raspberry:~ $ mosquitto_sub -h localhost -t test/topic
Bonjour le rapport
Bonjour le rapport
█
```

Dans ce terminal-ci, c'est l'abonné avec la commande **mosquitto\_sub**. Il s'abonne au topic test/topic sur le broker et affiche donc le contenu de celui-ci « Bonjour le rapport ».

Figure 26 & 27 : Test broker MQTT



Accéder au reste de la documentation : <https://juliannaapiomakani.be/>



Accéder au NutriTechPro : <https://nutritech.be/>