****

|  |
| --- |
|  |
| **Documentation du code pour le four**  Documentation et explication |
|  |

Page de service

**Niveau de confidentialité :** confidentiel

**Mises à jour**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Description du changement |
| 1.0.0 | 27/05/2023 | Paul BERNE | Création de document et explication des exigences |
| 1.1.0 | 06/06/2023 | Paul BERNE | Explications du code et du PDF |
| 2.0.0 | 19/06/2023 | Paul BERNE | Explications du code jusqu’à la fin |
|  |  |  |  |

Sommaire

[Page de service 0](#_Toc138683281)

[Sommaire 0](#_Toc138683282)

[1 Présentation générale du programme de communication entre l’ERP et le four thermidor 1](#_Toc138683283)

[1.1 Contexte 1](#_Toc138683284)

[1.2 Présentation de l’architecture logiciel 2](#_Toc138683285)

[1.3 Données nécessaires à l’exécution du programme 3](#_Toc138683286)

[1.4 Fonctionnement interne 3](#_Toc138683287)

[1.5 Documents fournis après traitement 4](#_Toc138683288)

[2 Détail et déroulé des sous-fonctions 5](#_Toc138683289)

[2.1 Mettre en place son environnement de travail 5](#_Toc138683290)

[2.2 Importations des modules Python 5](#_Toc138683291)

[2.3 Connexions a la base de données HELIOSII 6](#_Toc138683292)

[2.4 Analyse et traitement des données fournies par le four 6](#_Toc138683293)

[2.5 Tests d’intégration 17](#_Toc138683294)

[2.6 Comment utilisé le programme 18](#_Toc138683295)

[3 Informations complémentaires 19](#_Toc138683296)

[3.1 Problèmes de traitements des courbes 19](#_Toc138683297)

[3.2 Contacts 19](#_Toc138683298)

[4 Glossaire 20](#_Toc138683299)

[5 Bibliographie 20](#_Toc138683300)

[6 Table des illustrations 20](#_Toc138683301)

# Présentation de l’entreprise

## Fregate AERO

Fregate AERO a été créé en 2018 pour donner suite à la reprise de l’entreprise EFITAM qui avant été un sous-traitant aéronautique, puis en 2019 la filiale Fregate AERO SUD a été créé suite a la reprise d’AERO13 dans le Var.

Fregate AERO est une entreprise de production française, entretenant un savoir-faire de transformation de la matière qui exploite aux mieux les outils modernes, pour fabriquer des pièces aéronautiques, nous pouvons prendre l’exemple du train d’atterrissage pour hélicoptère qui est leurs produit phare.

# Présentation générale du programme de communication entre l’ERP et le four thermidor

## Contexte

Lors de la création de pièces certaines de celle-ci doivent être modifiées de formes pour cela une solution qui est de la chauffer a plusieurs centaines de degrés pour ensuite la manipuler, alors nous utilisons un four a haute température comme celui-ci :

Une image contenant acier, intérieur, ingénierie, Aluminium

Description générée automatiquement

Figure : Photo du four

Le client informe ensuite s’il veut que la pièce soit chauffée a tant de degrés pendant un certain temps et s’il veut que la pièce soit refroidie sois avec un bac de trempe :

Une image contenant acier, Matériau composite, bâtiment, escaliers

Description générée automatiquement

Figure : Bac de trempe

Dans un bac de trempe la pièce va être refroidi directement sans contrôle, contrairement au gradient ou la le client informe de combien la pièce doit perdre en degrés chaque heure pour un refroidissement contrôlé.

Pour lancer le four il faut que la personne chargée du four lance un programme qui correspond à plusieurs clients à partir d’un écran :

Une image contenant machine, intérieur, ingénierie, Atelier

Description générée automatiquement

Figure : Écran de contrôle du four

A savoir qu’un code est déjà déployé, c’est la version 5, mais cette version possède un problème majeur car elle affirme qu’une conformité avec les données exigées est bonne alors qu’elle ne l’est pas ce qui peut avoir des conséquences sur la pièce et l’entreprise car la pièce peut avoir été trop chauffé et donc elle ne sera pas efficace a 100% et cette pièce fera partie d’un avion donc nous n’avons pas le droit a l’erreur. Si un problème vient a se déclaré l’entreprise créatrice de la pièce seras accusé.

## Présentation de l’architecture logiciel

Voici le schéma ci-dessous de l’architecture logiciel pour que le programme fonctionne et dialogue avec les différents services :

Une image contenant capture d’écran, diagramme, ligne, texte

Description générée automatiquement

Figure : Infrastructure logiciel du programme

Explication du schéma :

Le PC ou est installé le programme, possède ODBC qui permet de faire le lien avec la base de données. A la fin du programme le même PC récupère sur l’ERP Hélios un PDF avec le graphe.

Le PC est connecté au réseau et sur ce réseau il récupère les fichier Group-1 et Group-2.

A partir de l’ODBC il se connecte a la base de donnée pour y récupérer toutes les données sur les clients.

Le programme envoie sur l’ERP le PDF du graphe.

Le Four communique avec un PC en lui envoyant les données de ses sondes et de toutes les données qu’il récupère, ce même PC est connecté au réseau et envoie toutes les données sous la forme d’un fichier .csv, fichier Group-1 et group-2.

## Données nécessaires à l’exécution du programme

Le programme de communication entre le four industriel et HELIOSII a pour objectif de **traiter des données de températures provenant du four, de vérifier leur conformité par rapport aux normes**, puis de générer des fichiers PDF qui contiennent ces données avec des graphiques, des [temps de maintien](#_Glossaire) et de transfert, ainsi que des informations sur les matières, les articles utilisés et le client. De ce fait, il est nécessaire que certaines données soient fournies avant l’exécution du programme pour que celui-ci puisse les traiter.

Deux documents sont lus par le programme dans le fichier courant :

* Un fichier *.csv* (Excel) de type « Groupe-1 » contenant les valeurs des températures au point chaud, point froid et régulation ainsi que les valeurs de puissance et la consigne de température. (cf. figure 4)
* Un fichier *.csv* (Excel) de type « Groupe-2 » contenant les valeurs de température du bac de trempe, le numéro de programme ainsi que les valeurs de puissance et la consigne de température. (cf. figure 5)

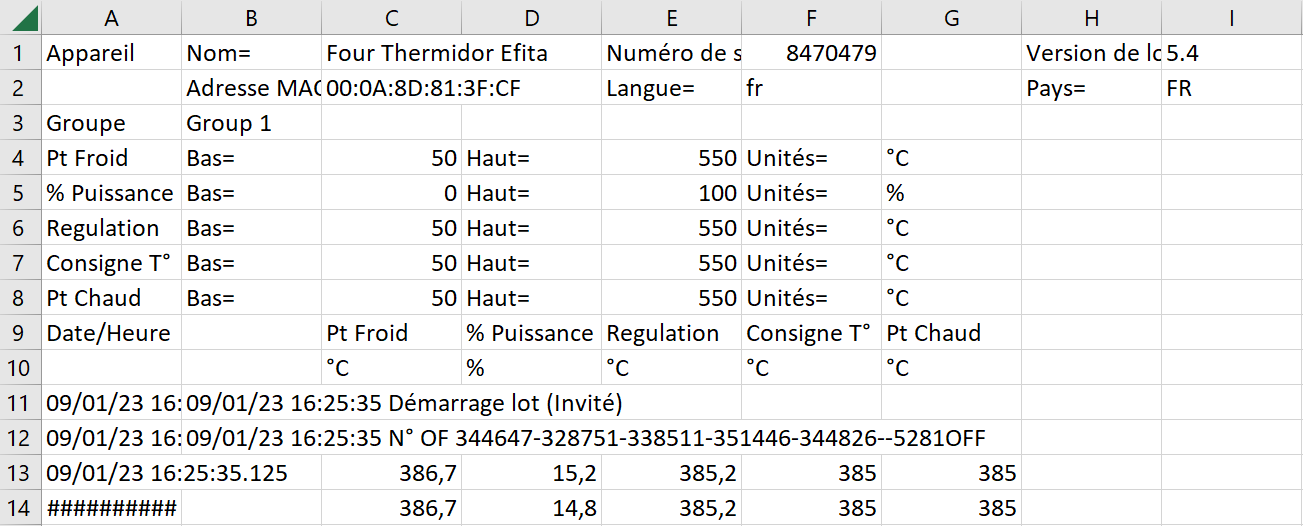
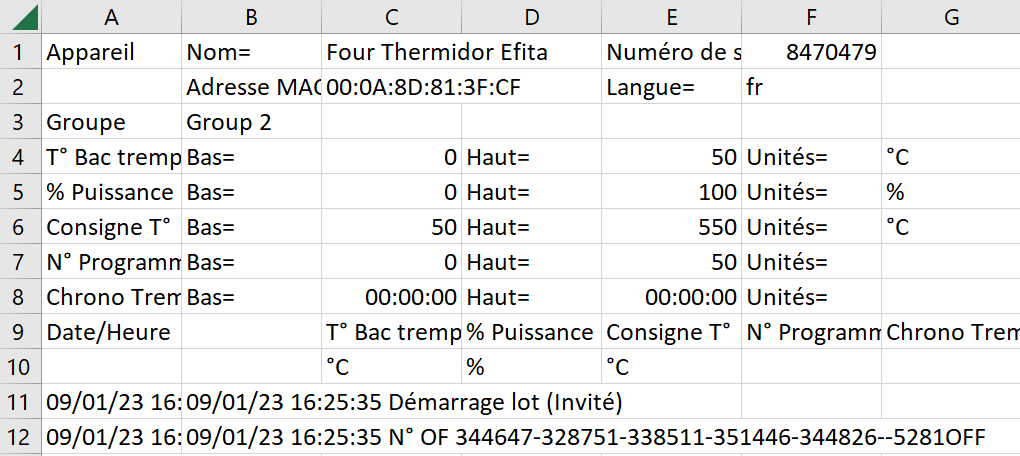


Figure : Exemple de document de type "Groupe-1"

Figure : Exemple de document de type "Groupe-2"



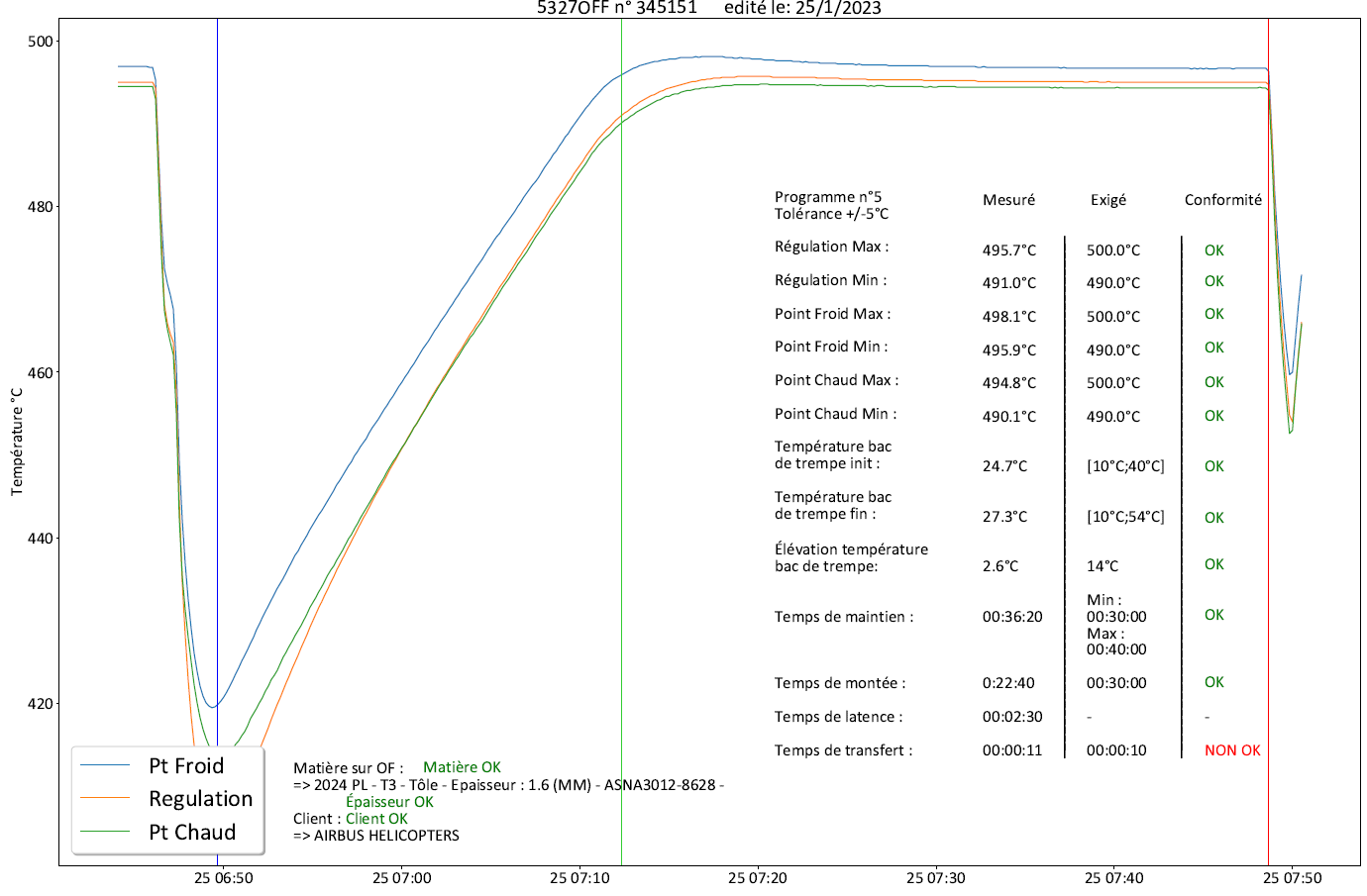
## Fonctionnement interne

Comme dit précédemment, le programme a pour but de fournir un fichier PDF graphique qui regroupe les informations de traitement fournies par le four Thermidor à la base de données HELIOSII. Les étapes d’exécution du code peuvent se résumer de la façon suivante :

* Importation des différentes bibliothèques utilisées pour se connecter à la base de données, lire les fichiers *.csv*, déplacer les fichiers traités dans un autre répertoire, tracer des graphiques et enregistrer des informations dans un fichier de log.
* Connexion à la base de données Oracle nommée "HELIOSII", récupération des données clients (réf clients, exigences).
* Lecture des fichiers *.csv* dans le répertoire courant et traitement des données contenues dans les fichiers.
* Vérification si un fichier IMP 650 existe car il contient toutes les données nécessaires pour réaliser la comparaison des données récupérées a l’exigence client. Ensuite il nous faut vérifier s’il y’a un ou plusieurs et il faut qu’il ait un fichier group-2 qui lui correspond ou sinon il n’est pas déplacé dans le fichier automoved
* Une fois les données des fichiers .csv récupérées, le programme vérifie que les valeurs sont conformes aux normes fournies par le(s) client(s) et valide ou non les différentes informations du traitement thermique dans la colonne « Conformité ».
* Le script crée un fichier PDF à partir du graphique de températures et des données fournies par le four. Puis, il l'enregistre dans HELIOSII en liant le fichier à l’OF de la pièce.

## Documents fournis après traitement

Une fois les données traitées, le programme les met en forme sur un fichier PDF de la façon suivante :



**Données du traitement appliqué à la fournée**

**Données du traitement prévu par le programme du four**

Figure 3 : Exemple de graphique fourni par le programme après exécution

Les courbes verticales sont importantes dans le calcul des temps de montée et de maintien.

En effet, le [temps de montée](#_Glossaire) se calcule par soustraction du temps où commence le programme du four (droite verte) et de l’instant de la fermeture de la trappe (droite bleu). Le raisonnement est le même pour le calcul [des temps de maintien](#_Glossaire), mais la soustraction se fait entre les droites rouge et verte. Elles sont aussi présentes sur le graphique pour vérifier visuellement que les [temps de montée](#_Glossaire) et de maintien sont cohérents, une droite placée au mauvais endroit entraine nécessairement un calcul faux.

Sur le fichier, sont affichés les valeurs minimum et maximum des trois sondes de température (point froid, point chaud et régulation) dans l’intervalle du [temps de maintien](#_Glossaire).

Dans le cas d’une trempe, le fichier comportera aussi les valeurs de température du bac de trempe avant et après la trempe ainsi que l’élévation en température que le bac à subit au cours de cette dernière.

[Les temps de maintien](#_Glossaire) (temps compris entre l’instant où la dernière sonde entre dans l’exigence de température et l’instant où la première sonde sors de cette exigence), [le temps de latence](#_Glossaire) (temps compris entre l’instant où la première sonde entre dans l’exigence de température et l’instant où la dernière sonde entre dans cette exigence) et [le temps de transfert](#_Glossaire) dans le cas d’une trempe (temps compris entre l’ouverture des portes et l’immersion totale du panier dans le bac de trempe).

# Détail et déroulé des sous-fonctions

## Mettre en place son environnement de travail

Pour pouvoir utiliser le programme sans problème il faut premièrement mettre ne place son environnement de travail.

Pour cela veuillez installer python avec ce lien <https://www.python.org/downloads/> , puis lors de l’installation en lançant l’exécutable téléchargé veuillez bien cocher la case créé la variable d’environnement.

Désormais nous pouvons installer tous les modules nécessaires :

Veuillez pour cela faire ses commandes l’une après l’autre :

**pip install os glob csv shutil math logging pyodbc numpy openpyxl simple\_colors matplotlib dateutil time**

## Importations des modules Python

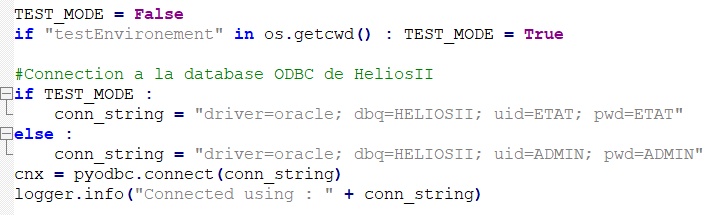
Les premières lignes du programme commencent par l’importation des modules et librairies qui vont être utilisées dans la suite du code, en voici un résumé :

* *os* = Le module os est un module fournit par Python dont le but **d'interagir avec le système d'exploitation**, il permet ainsi de gérer l’arborescence des fichiers, de fournir des informations sur le système d'exploitation processus, variables systèmes, ainsi que de nombreuses fonctionnalités système...
* *glob* = Le module glob en Python est utilisé pour trouver tous les noms de chemins qui correspondent à un modèle spécifié selon les règles utilisées par la ligne de commande Unix. Il offre un moyen pratique de **rechercher des fichiers et des répertoires** en utilisant des caractères génériques.
* *csv* = Permet de **lire et écrire des données tabulaires** au format *.csv*.
* *shutil* = Le module shutiloffre un certain nombre d'opérations de haut niveau sur les fichiers et les collections de fichiers. En particulier, des fonctions sont fournies qui prennent en charge la **copie et la suppression de fichiers**. Pour les opérations sur des fichiers individuels, voir aussi le module os.
* *math* = Fournit l’**accès aux fonctions mathématiques** définies par le standard C.
* *logging* = Recueille dans un fichier **toutes les informations importantes sur les incidents de fonctionnement**. Logger enregistre **les actions pendant que le programme fonctionne**. Il n'apparaît pas directement en tant qu'instance : il faut l'appeler avec la fonction *logging.getLogger(Loggername).*
* *Sys* = Fournit des fonctions et des variables qui permettent d’**interagir avec l’interpréteur Python** telles que la version de l’interpréteur, la valeur maximale que la variable peut contenir, les informations de copyright de l’interpréteur, etc. Il nous donne des informations sur les éléments, les fonctions et les méthodes de l’interpréteur. Lorsqu'une exception est levée et n'est pas interceptée, l'interpréteur appelle *sys.excepthook* avec trois arguments, la classe de l'exception, l'instance de l'exception, et un objet traceback.
* *pyodbc* = Module open source qui simplifie l'**accès aux bases de données ODBC** (ici à « HELIOSII »).
* *matplotlib.pyplot* = Regroupe un grand nombre de fonctions qui servent à **créer des graphiques et à les personnaliser** (travailler sur les axes, le type de graphique, sa forme et rajouter du texte).
* *numpy* = La bibliothèque numpy est utilisée dans le programme pour mettre en forme des données dans un tableau de type *np.array()* et ainsi faciliter leur traitement.

## Connexions a la base de données HELIOSII

Une fois toutes les librairies importées, le programme effectue une connexion à la base de données HELIOSII, soit en tant qu’ADMIN soit en tant qu’ETAT.

**À noter** que la connexion en tant qu’ETAT se fait seulement si le programme est exécuté en TEST\_MODE. Le TEST\_MODE s’active si et seulement si le nom du dossier dans lequel se trouve le programme contient « testEnvironnement ».

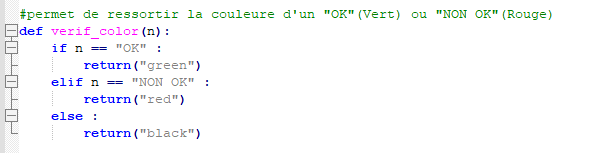


Une fois ces lignes de code exécutées, la connexion avec HELIOSII est alors effectuée.

## Analyse et traitement des données fournies par le four

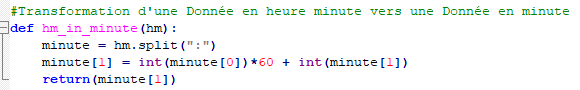
Comme énoncé précédemment, le programme récupère les données des fichiers .csv et les utilises pour pouvoir tracer les courbes de températures. Cependant, il transmet plusieurs autres informations à différents fichiers, il est alors préférable de détailler ce que fait chaque fonction du programme.

**La fonction verif\_color :**

****

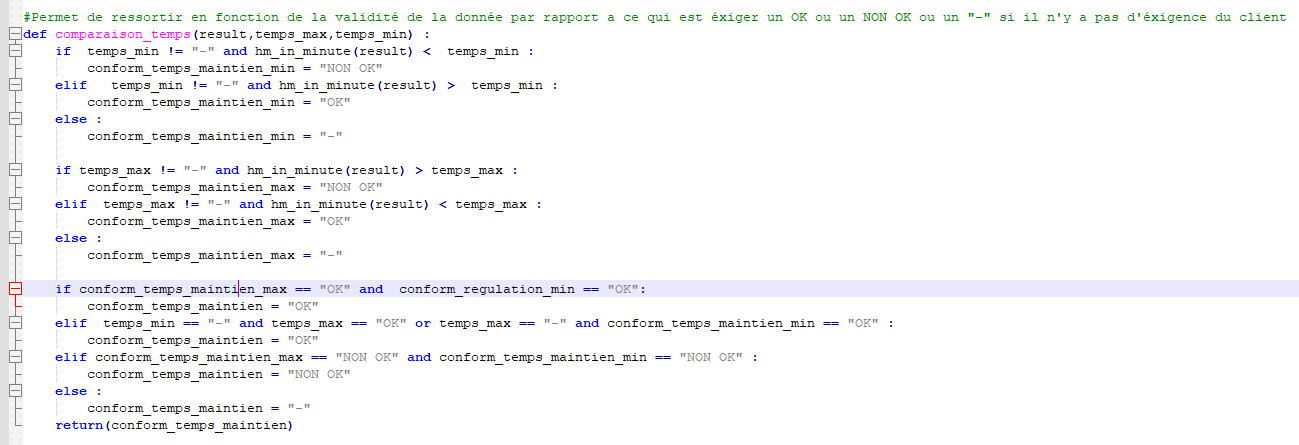
Cette fonction permet lors de la vérification faite entre les données récoltées et celles exigées de les comparées et d’affiché en fonction de la conformité le texte en vert si c’est conforme et rouge si ça ne l’est pas.

**La fonction hm\_in\_minute :**



Cette fonction est simple elle sert simplement à convertir un format d’heure : minute en simple minutes totales.

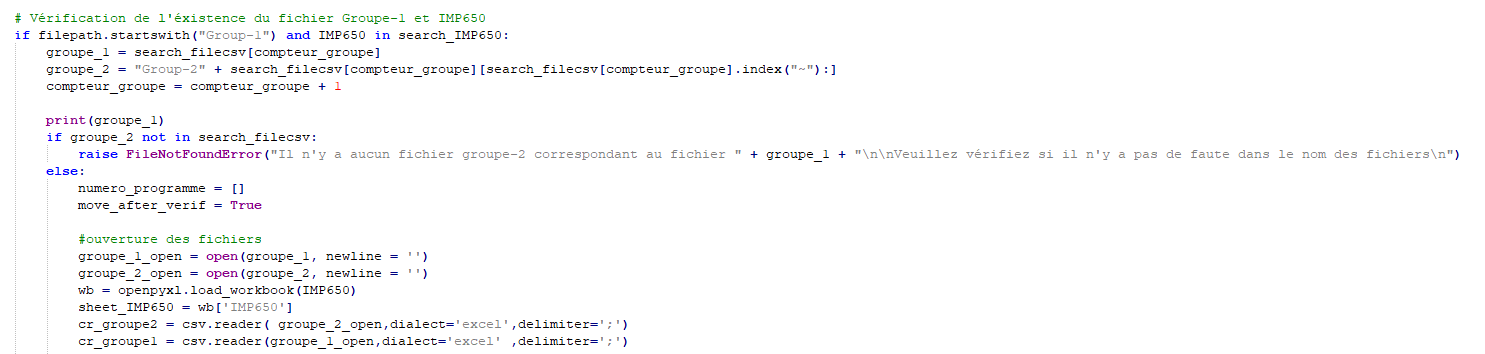
**La fonction comparaison\_temps :**



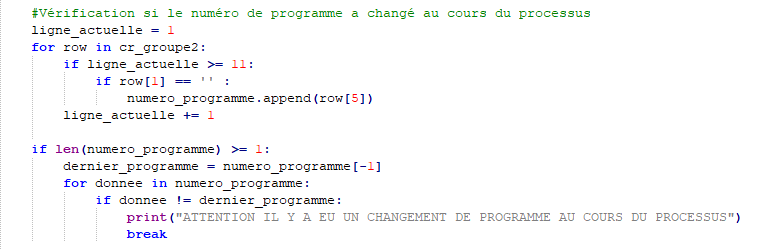
Comme marqué dans le nom cette fonction permet de comparé des temps récupéré au temps exigé pour en ressortir un NON OK ou un OK ou un – s’il n’y a pas d’exigence, sur le PDF.

**La boucle for qui est de la ligne 100 a la ligne 699 permet de parcourir chaque fichier .csv du répertoire actuel :**

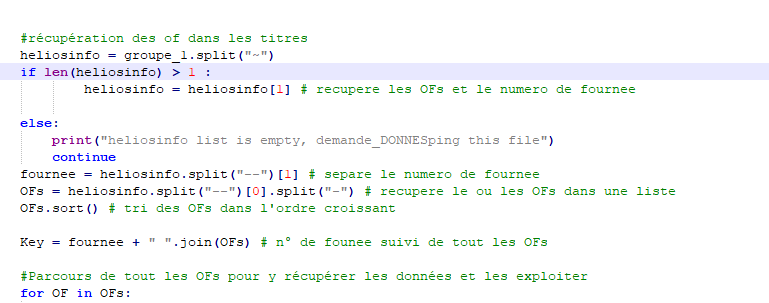
Ensuite il y’a la partie sur toute la vérification de l’existence de tous les fichiers nécessaires pour faire tout la procédure :



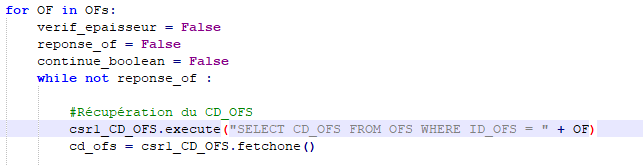
Ensuite il y’a la partie où nous récupérons la donnée qui principale car c’est elle qui nous guide, car c’est le numéro de programme et permet de chercher les données exigées même si ce n’est pas la seule donnée :



Ensuite il y a la partie ou nous récupérons les OFS dans les titres du fichiers groupe-1, un OF est un ordre de fabrication et correspond a un produit et un client, et nous les parcourons 1 par 1 :



Puis pour trouver toutes les données nécessaires pour faire la comparaison avec ce qui est exigé dans l’IMP650 il nous faut le numéro de programme le client la matière et l’épaisseur donc toutes ces données mise à part le numéro de programme nous allons les retrouvées dans la base de données voici la requête SQL :



Avec la récupération de la clé primaire cd\_ofs à partir de l’id of contenu dans le titre

WITH EPAISSEUR\_MATIERE AS (

SELECT

o.CD\_ARTICLE,

CASE

WHEN INSTR(REF\_LIBELLE, 'Epaisseur') <> 0 THEN SUBSTR(REF\_LIBELLE, INSTR(REF\_LIBELLE, 'Epaisseur') + 12, INSTR(REF\_LIBELLE, '(MM)', INSTR(REF\_LIBELLE, 'Epaisseur')) - INSTR(REF\_LIBELLE, 'Epaisseur') - 13)

ELSE NULL

END AS Epaisseur,

SUBSTR(REF\_LIBELLE, 1, INSTR(REF\_LIBELLE, ' - ') - 1) AS Matiere

FROM

OFS\_BESOIN ob

INNER JOIN OFS o ON o.CD\_OFS = ob.CD\_OFS AND TRIM(ID\_OFS) = " + str(OF) + "

)

SELECT DISTINCT

em.Epaisseur,

em.Matiere,

c.NOM

FROM

Client c

INNER JOIN Article a ON a.CD\_CLIENT = c.CD\_CLIENT

INNER JOIN EPAISSEUR\_MATIERE em ON a.CD\_ARTICLE = em.CD\_ARTICLE

ORDER BY

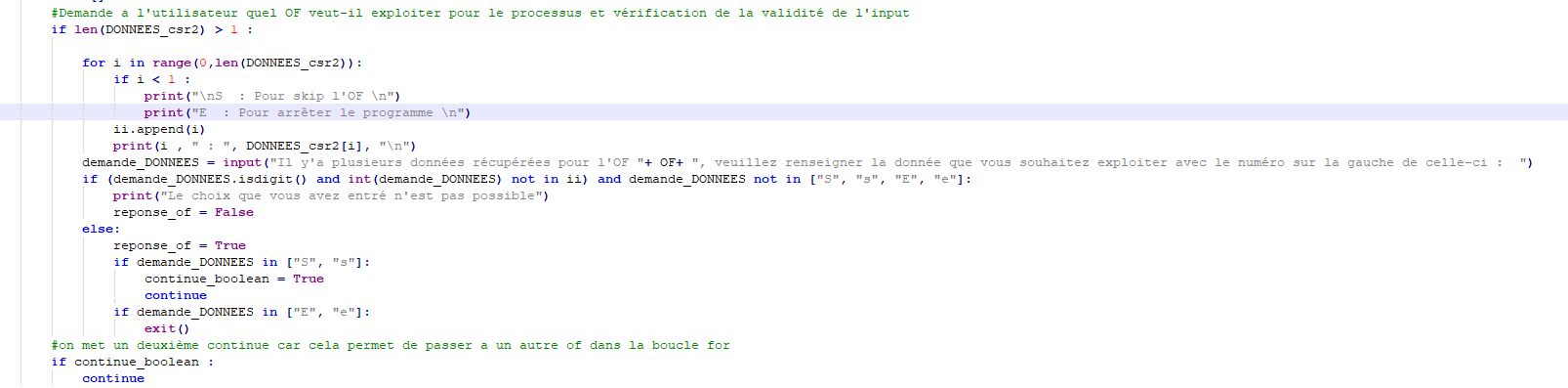
c.NOM,

em.Matiere,

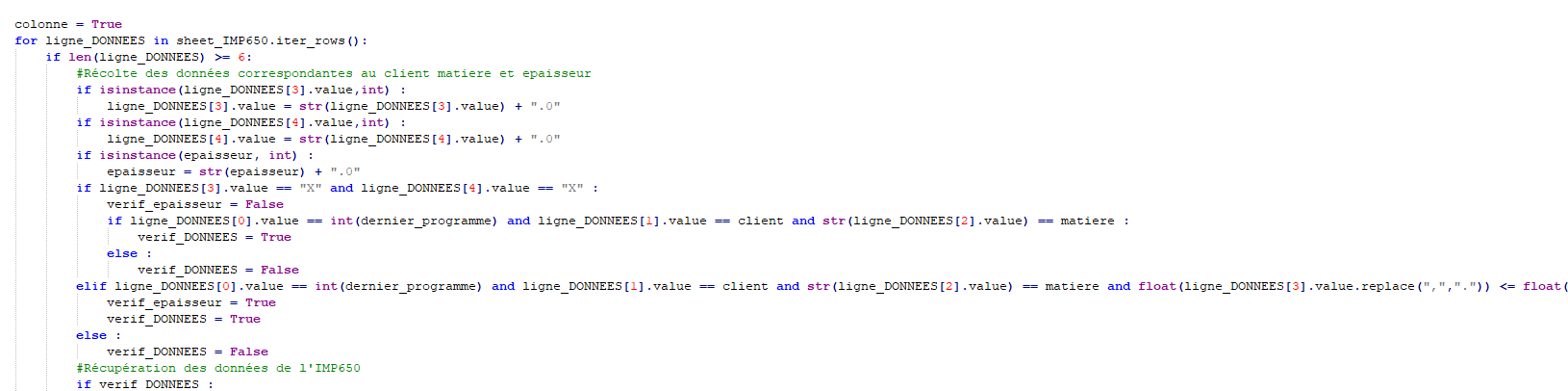
em.Epaisseur

Dans cette requête nous allons dans la table article pour y retrouver la matière et l’épaisseur dans la même colonne ensuite nous découpons le texte pour y ressortir l’épaisseur dans une colonne et la matière dans une colonne puis nous récupérons le client dans la table client.

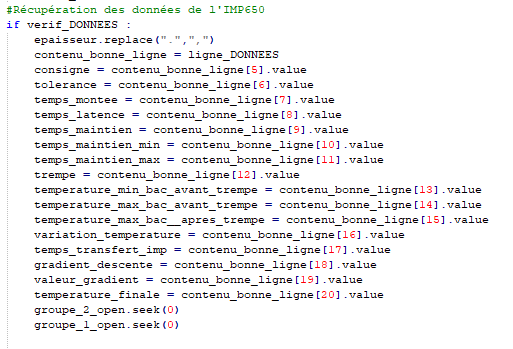
Ensuite dans la partie d’après nous demandons a l’utilisateur quelle données il souhaite utilisé lors du processus car il est possible que l’on ressorte plusieurs données suite à la requête SQL :



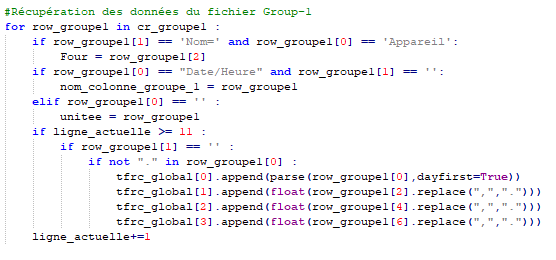
Sur la partie d’après nous prenons les données récupérées pour aller dans le fichier IMP650 et trouver la ligne qui correspond à toutes nos données :



Puis après avoir trouvé la ligne correspondante nous récoltons ces données qui vont donc correspondre a ce qui est exigé par le client :



Puis nous récupérons les données du fichier groupe-1 et plus particulièrement nous récupérons :



La sonde point chaud

La sonde régulation

La sonde point froid

Le temps

Le nom des colonnes

Le nom du four

Ensuite nous découpons le fichier groupe-1 en quatre parties :

La redescente qui est le début des données, mais aussi lorsque les données redescendent, nous mettons un index a la fin de cette période :



La montée qui commence lorsque la redescente a fini et que les données commence a remontée, et la montée finis lorsqu’une sonde a atteint la consigne minimale :



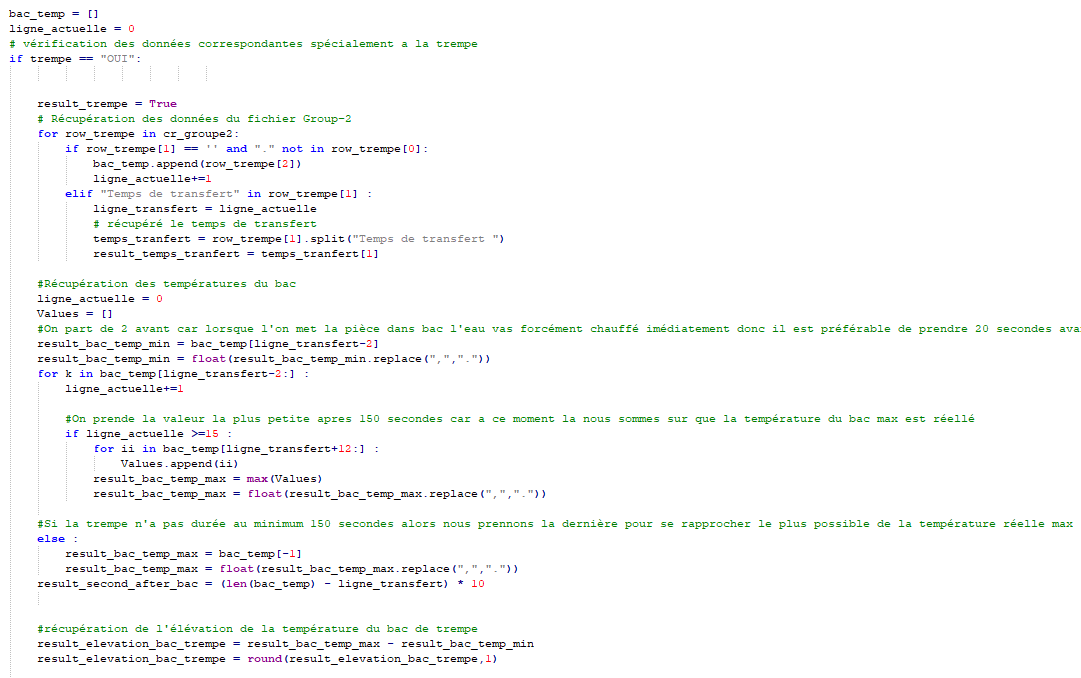
C’est à ce moment-là que la latence commence, lorsqu’une sonde a atteint la consigne minimale et se finis lorsque les trois sondes sont dans la consigne minimale :



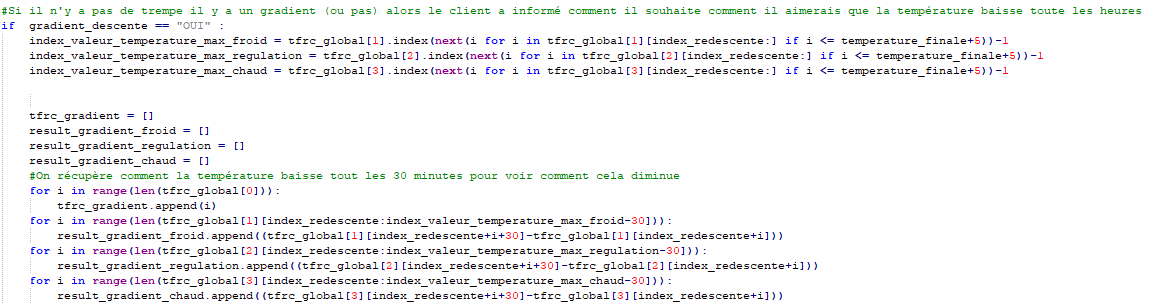
Cela marque donc le début du maintien, qui est la phase lorsque les sondes sont toutes les 3 dans l’intervalle et ce finis lorsque en partant de la fin la première sonde qui sort de la consigne minimale :



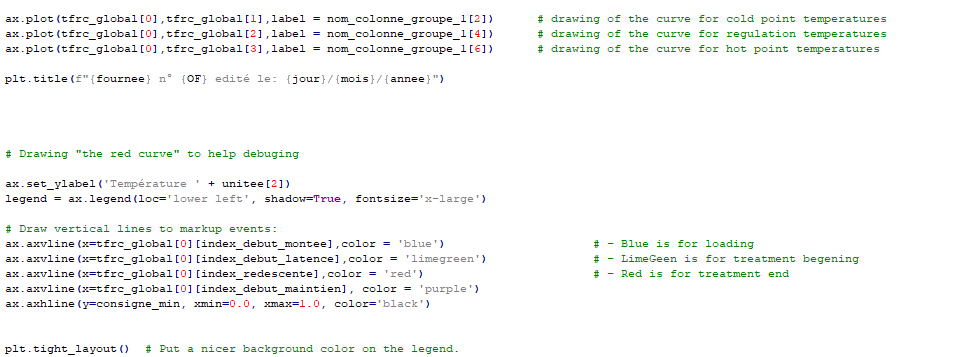
Dans la prochaine partie du code nous effectuons tout ce qui est nécessaire dans la récolte des données s’il y’a une trempe suite au processus du four, ce qui nous intéresse est dans le fichier groupe-2. Nous récoltons la température du bac initiale, la température maximale et le temps qu’il s’est écoulé lorsque la pièce était dans le bac de trempe, puis nous regardons le temps de transfert qui le temps du moment où la pièce est sortie du four pour allez dans le bac de trempe :



Dans la partie suivante nous faisons les mêmes récoltes de données mais pour le cas ou il y’a un gradient, avec les informations qui nous intéresse qui sont la valeur du gradient pour la sonde froide, régulation, et point chaud.

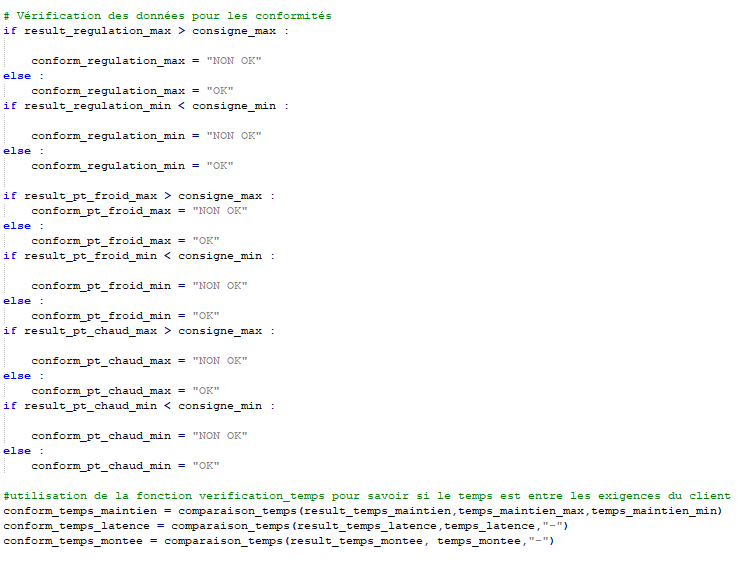


Dans la partie qui suit nous commençons à faire le traitement sur notre fichier PDF ou là nous traçons la courbe des données et des droite vertical aux index du début de montée, début de la latence, au début du temps de maintien, au début de la redescente, et une ligne horizontale au niveau de la consigne minimale :

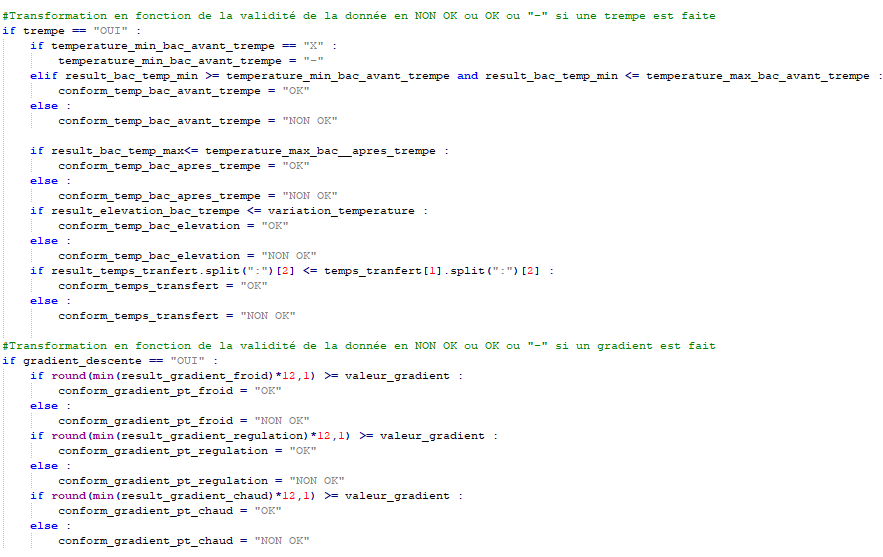


Ensuite de la ligne 453 a la ligne 508 ce n’est que simplement des récoltes de données pour ensuite les comparés.

C’est ce qu’il se passe dans le bloc suivant, ou là nous réalisons la comparaison des données récolées à l’exigence du client :



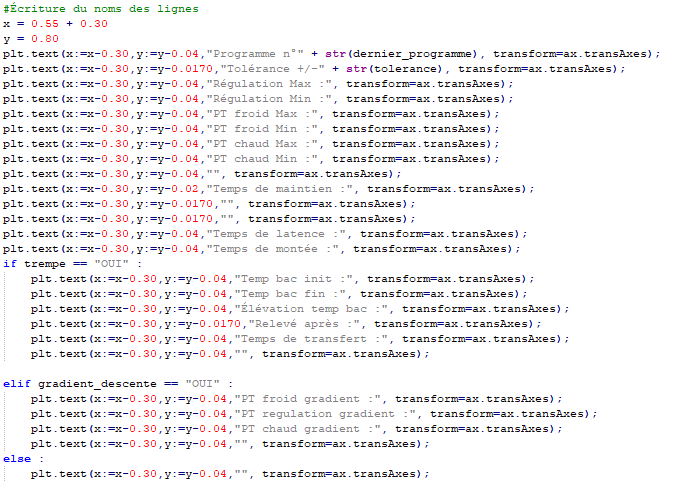
Nous faisons les vérifications différentes s’il y a une trempe ou un gradient :



Désormais nous avons les principales données que nous avons besoin pour afficher sur le PDF les conformités, il ne suffit plus qu’a écrire sur le PDF :

Sur le code l’écriture du fichier est répartie sur 3 blocs :

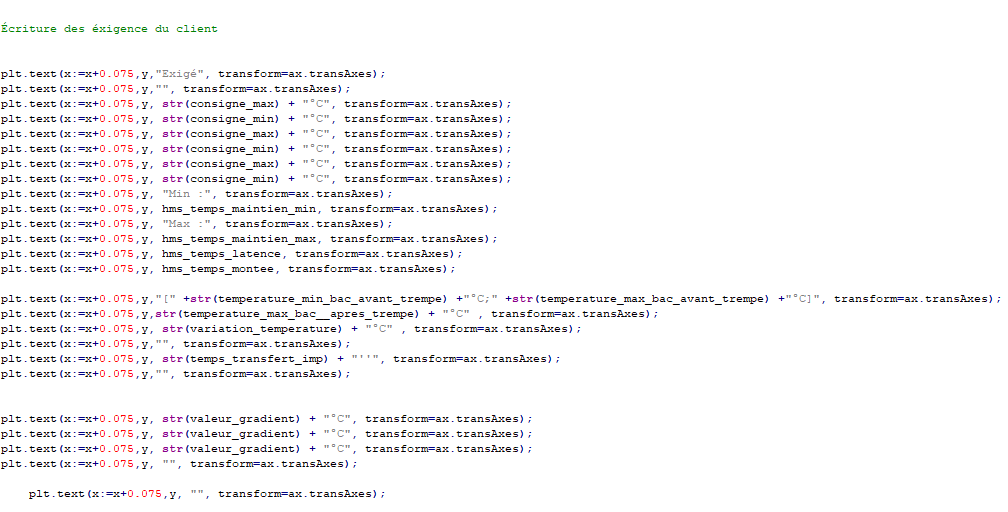
**Les noms de lignes :**

****

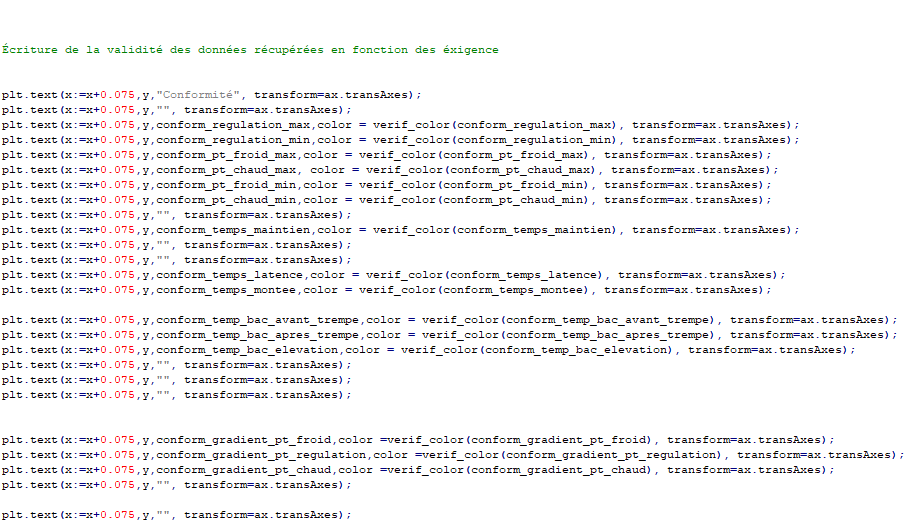
**Les données récupérées :**

****

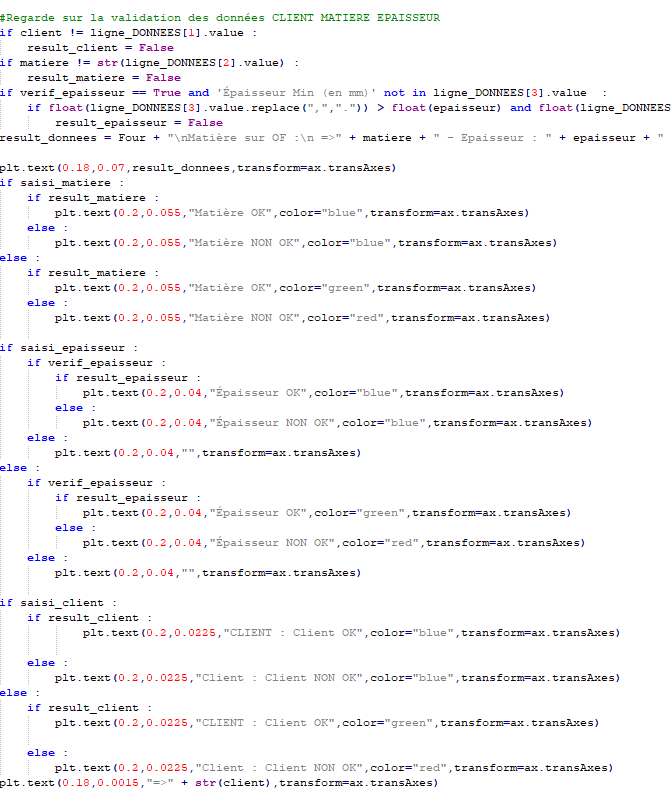
**Les données exigées :**

****

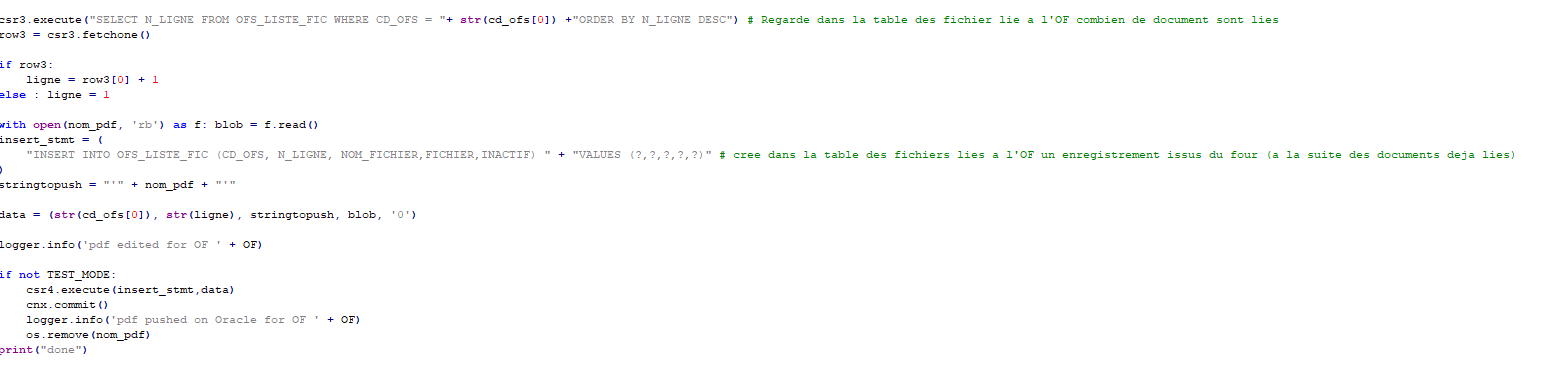
**Si les données sont conformes ou pas :**



Ensuite sur le bloc suivant nous affichons simplement les données qui ont été récupérés au début du processus donc le client, la matière et l’épaisseur, et nous les affichons en bas à gauche du PDF, et si une donnée a été informé à la main elle s’afficheras en bleu pour qu’il n’y a pas de problème dans la compréhension :



Et pour le bloc final nous insérons le PDF dans l’ERP HELIOS :



## Tests d’intégration

Nous nous sommes heurtés à un problème sur une courbe car pour mettre une ligne verticale a la redescente nous prenons dernière valeur, en partant de la fin, qui ne correspond plus a la consigne minimale (valeur souhaitée – tolérance). Mais pour une courbe, le programme s’arrête mais au mauvais endroit, car a la fin du fichier les données repasse au-dessus de la consigne minimale et donc pour le programme le processus s’arrête a cet endroit. (cf :Figure 6)

Une image contenant texte, diagramme, ligne, Tracé

Description générée automatiquement

Trait exigé

Trait faussé

Figure 7 : PDF du test d'intégration

Donc pour résoudre ce problème nous pouvons trouver une autre solution pour récupérer quand le [temps de redescente](#_Glossaire) commence.

Pour cela nous pouvons simplement partir de la fin et de trouver la dernière sonde qui est en dessous de la consigne minimale et qui a pour donnée d’après une donnée au-dessus de la consigne minimale et voici le résultat de la ligne de commande :

index\_redescente = min(len(tfrc\_global[1]) - 1 - list(reversed(tfrc\_global[1])).index(next(i for count, i in enumerate(list(reversed(tfrc\_global[1][index\_debut\_maintien:]))) if i < consigne\_min and list(reversed(tfrc\_global[1][index\_debut\_maintien:]))[count+1] >= consigne\_min))+1,len(tfrc\_global[2]) - 1 - list(reversed(tfrc\_global[2])).index(next(i for count, i in enumerate(list(reversed(tfrc\_global[2][index\_debut\_maintien:]))) if i < consigne\_min and list(reversed(tfrc\_global[2][index\_debut\_maintien:]))[count+1] >= consigne\_min))+1,len(tfrc\_global[3]) - 1 - list(reversed(tfrc\_global[3])).index(next(i for count, i in enumerate(list(reversed(tfrc\_global[3][index\_debut\_maintien:]))) if i < consigne\_min and list(reversed(tfrc\_global[3][index\_debut\_maintien:]))[count+1] >= consigne\_min))-1)

Résultat :

Une image contenant texte, diagramme, ligne, Tracé

Description générée automatiquement

## Comment utilisé le programme

Premièrement veuillez bien faire attention qu’il y est les fichier groupe-1 et group-2 correspondant dans le répertoire du programme.

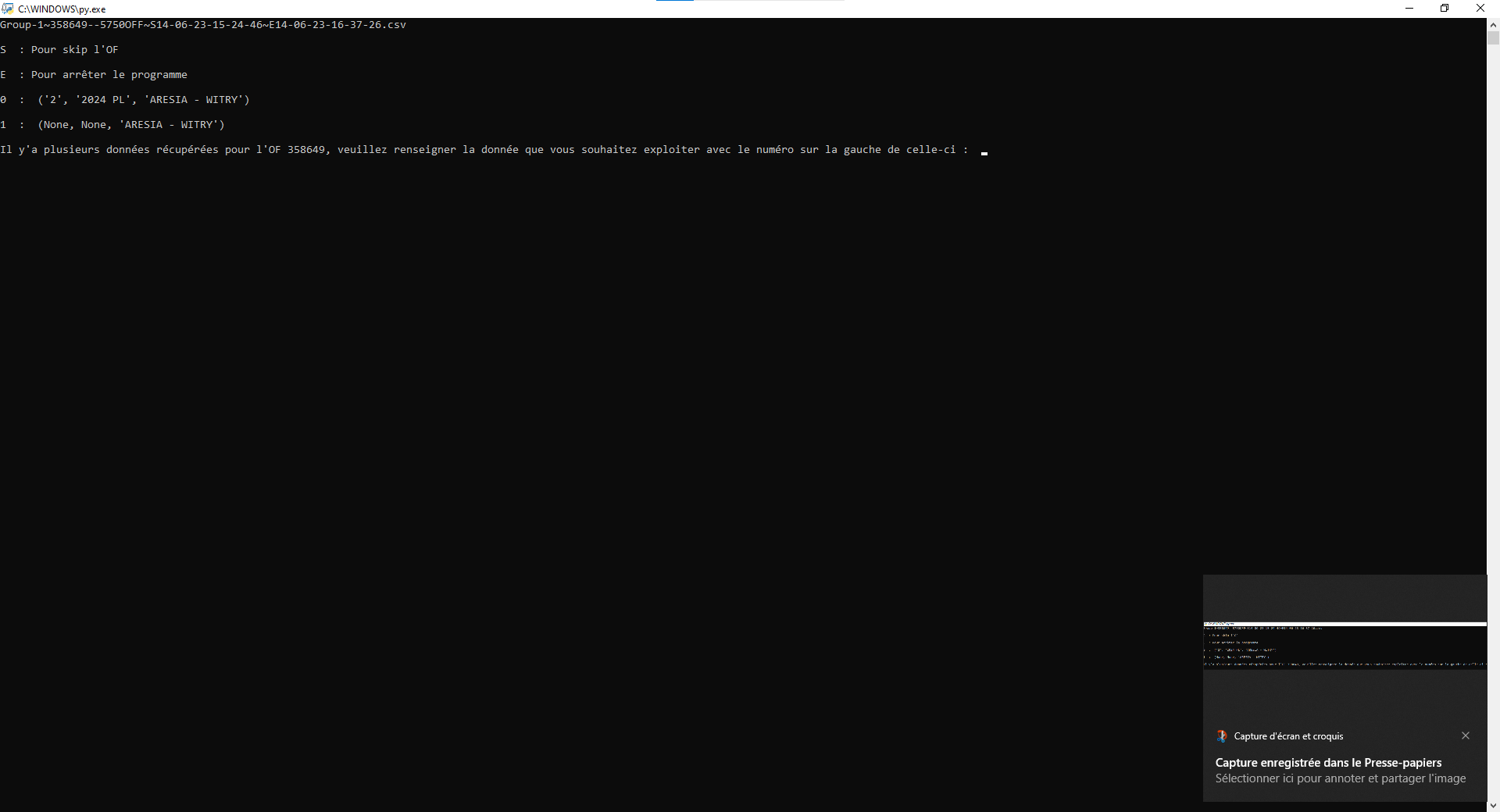
Ensuite faite attention que dans le chemin du répertoire il n’y est pas le nom « testenvironnement ».

Veuillez ne pas avoir d’ouvert les PDF correspondant à un OF, et veuillez ne pas ouvrir les fichier group-1, group-2 et IMP650 lorsque vous lancez le programme.

Et pour finir veuillez au préalable avoir installé tous les modules présentés au début du document.

Vous pouvez lancer le programme, voici comment se déroule le programme :

Tout d’abord double cliquer sur le programme et attendais jusqu’à voir apparaître un texte vous demandant ceci :



Ensuite ici vous avez plusieurs choix :

Sois l’OF présenté n’est pas celui souhaité ou autre vous peut le passer pour allez au suivant et faire le traitement, alors il vous suffit de marquer « s ».

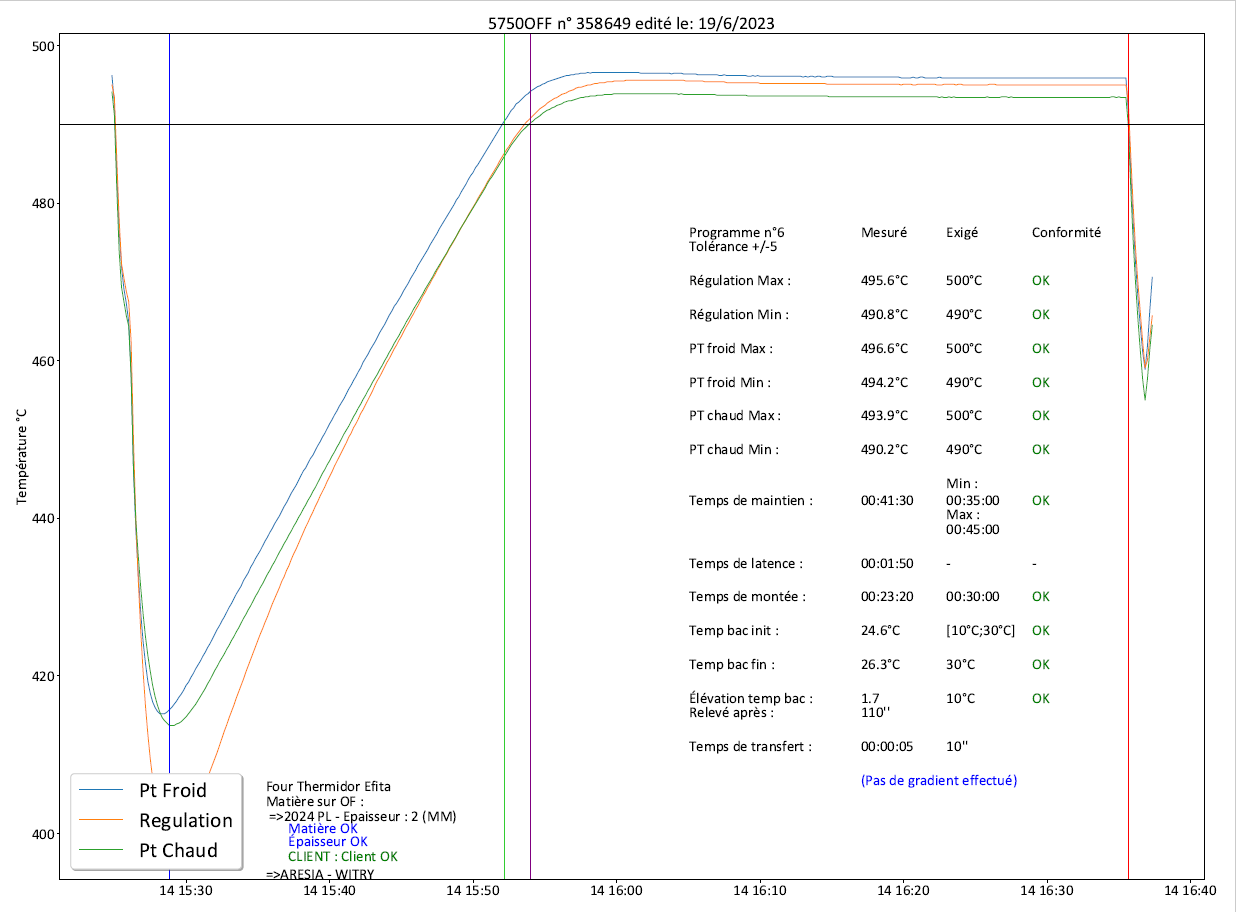
Sois vous avez lancé avec le mauvais fichier et vous ne souhaité pas ou plus faire le traitement alors il vous suffit de marquer « e ».

Soit-il s’agit du bon OF avec les bonnes informations et vous avez a tapé le numéro correspondant aux données souhaitées.

Dans notre cas nous allons choisir l’option 1 :

Et nous entrons les données souhaitées.

Voici le PDF qui en ressors, en ouvrant le PDF nommé TTH\_ « le nom de fournée » \_ « le nom de l’OF ».pdf :



Avec en bleu en bas à gauche les données entrées à la main

# Informations complémentaires

## Problèmes de traitements des courbes

Il peut arriver que certaines fournées ne soient pas traitées par le programme. Ce problème provient généralement d’une donnée manquante dans les fichiers .*csv* fournis par le four. À la fin de chaque traitement thermique, une ligne « Arrêt lot » est créée à la fin du fichier.

Cependant, il peut arriver que l’enregistreur du four ne détecte pas la fin du programme et donc n’inscrit pas cette ligne dans le tableau. C’est pourquoi il est important de vérifier lorsqu’une fournée ne peut être traitée par le programme, que l’absence de la ligne « Arrêt lot » n’est pas la cause du problème.

## Contacts

Cette notice d’utilisation a été réalisée par Paul BERNE auteur de la version 6 du programme de traitement thermique du four.

Pour toute information merci de me contacter à l’adresse suivante : paulberne1309@gmail.com

# Glossaire

Trempe = C’est le fait de mettre la pièce après le processus de réchauffement dans un bac rempli d’eau

Gradient = C’est le fait de contrôlé le refroidissement de la pièce après le processus

Temps de latence = Temps qui démarre lorsqu’une sonde passe dabs la consigne minimale et qui fini lorsque la dernière sonde rentre dans la consigne minimale.

Temps de maintien = Temps qui démarre lorsque les 3 sondes sont supérieures à la consigne minimale et se finis lorsqu’une sonde n’est plus dans la consigne minimale.

Temps de montée = Temps qui commence lorsque dans le premier cinquième du temps les données commence a augmentées et il se finis lorsqu’une sonde est plus grande que la consigne minimale.

Temps de redescente = Temps qui commence lorsque la première sonde sort de la consigne minimale en partant du temps de maintien jusqu’à la fin, et se finis à la fin du fichier.

Temps de transfert = Est lorsqu’une trempe doit être faite, c’est le temps entre lequel la pièce passe du four au bac de trempe

# Table des illustrations

[Figure 1 : Photo du four 1](#_Toc138686880)

[Figure 2 : Bac de trempe 2](#_Toc138686881)

[Figure 3 : Écran de contrôle du four 2](#_Toc138686882)

[Figure 4 : Infrastructure logiciel du programme 3](#_Toc138686883)

[Figure 5 : Exemple de document de type "Groupe-1" 4](file:///E:\entreprise\stage%201ereannee%20freegate\document%20dans%20le%20stage\mission2\doc\documentation_code_Four.docx#_Toc138686884)

[Figure 6 : Exemple de document de type "Groupe-2" 4](file:///E:\entreprise\stage%201ereannee%20freegate\document%20dans%20le%20stage\mission2\doc\documentation_code_Four.docx#_Toc138686885)

[Figure 7 : PDF du test d'intégration 18](#_Toc138686886)