République Tunisienne

Ministère de l’Enseignement Supérieur

et de la Recherche Scientifique

Université de Sfax



Ingénieur en :

Génie informatique industrielle

École nationale d’électronique

et des télécommunications de Sfax

**RAPPORT DE STAGE**

*Présenté à :*

**L’école Nationale d’Electronique et des Télécommunication**

*Filière :*

**Diplôme National d’Ingénieur en :**

**Génie Informatique Industrielle**

*Par :*

**Yassin zghal**

Ajustement du script d’automatisation de test unitaire

ORGANISME D’ACCEUIL :

**ACTIA Engineering Services**



PERIODE DU STAGE :

**2/7/24 – 15/8/24**

REMERCIEMENTS :

C'est avec une immense joie que je saisis l'opportunité de témoigner ma profonde

gratitude envers tous ceux qui m'ont soutenu et accordé leur confiance tout au long de mon

stage au sein de la société **ACTIA Engineering Services**. En premier lieu, je tiens à exprimer

mes sincères remerciements à mon directeur des études et des stages à l'**ENET'COM**, ainsi

qu'à **tous les cadres de ACTIA Engineering Services** qui m'ont offert l'opportunité

d'effectuer ce stage de découverte professionnelle.

J'adresse également mes plus chaleureux remerciements à **Mme Latifa SAIDI** pour sa

bienveillance et son précieux soutien tout au long de ce stage. Je suis profondément

reconnaissant envers tous ceux qui n'ont jamais hésité à me guider, à me fournir de la

documentation, et à me montrer les diverses facettes de leur travail. Leur aide inestimable a

grandement contribué au succès de mon stage, en me permettant de le réaliser dans les

meilleures conditions possibles.

Cette expérience a non seulement enrichi mes connaissances techniques, mais m'a

également offert une perspective plus claire de mon futur professionnel . Merci infiniment

pour tout ce que vous avez fait pour moi.



*CHAPITRE 1 : Présentation l’entreprise ACTIA Engineering Services*

*I. Introduction générale de l’entreprise : …………………………………………………………… 6*

*II. Les Compétences et Expertises De ACTIA : ……………………………………………………* ***7***

*II-1. Développement de Logiciels Embarqués : …………………………………………………7*

*II-2. Développement Web et Mobile : ……………………………………………………………. 7*

*II-3. Développement Électronique : ……………………………………………………………… 8*

*II-4. Réalisation de Tests et Validations Produits : …………………………………………… 8*

*III. Introduction Générale sur le Domaine du Test Unitaire : ……………………………………9*

*IV. Les outils utiliser en Test Unitaire :……………………………………………………………. 9*

*Chapitre 2 : Cadre du projet*

*I. Introduction : ………………………………………………………………………………………11*

*II. Objectifs du Projet : …………………………………………………………………………… 11*

*III. Les étapes du Projet : …………………………………………………….……………………11*

*IV. Contexte du Projet : …………………………………………………………….……………. 12*

*IV. 1- Analyse du Script .bat(script\_generation) : ……………………………………………12*

*IV.1.1- Préparation de l'environnement : ………………………….……………………… 13*

*IV.1.2- Exécution des tests TBrun : …………………………………….……………………13*

*IV.1.3- / Exécution de LDRA : …………………………………….…………………………14*

*IV.1.4- Exécution des fichiers TCF : …………………………….…………………………. 14*

*IV.1.5- Extraction et copie des rapports : ………………..……….………………..………15*

*IV.2- Analyse du Script Python (script\_synthese) : ……………….………………………...17*

*IV.2.1- Le résultat final : ………………………………………….…………………………18*

*V. Problématique : …………………………………………………….……………….…………19*

*Chapitre 3 : Exécution du Projet*

*I. Conception du Nouveau Script python : ………………………………………………………20*

*II. explication des importations utilisée : ………………………………………………………………… 20*

*III. Transformer le code .bat en code Python : ……………………………………………………………21*

*III.2- Explication des fonctions : …………………………………………………………………………21*

*IV. Amélioration du code (script\_synthese) : ……………………………………………………………… 23*

*IV.1- Explication des fonctions : …………………………………………………………………………23*

*V. les modifications réalisées de code final : ………………………………………………………………24*

*V.1- Affichage de l'Heure de Début et de Fin du Script : ………………………………………………25*

*V.2- Gestion des Erreurs : ………………………………………………………………………………… 25*

*V.3-Documentation et Maintenance : …………………………………………………………………… 26*

*VI. convertir le code python .py en un exécutable .exe : …………………………………………………26*

*VII. Conclusion : ………………………………………………………………………………………………27*



Figure 1 : image se software LDRA………………………………………………………………………….. 9

Figure 2**:** Sélection de l’applications LDRAunit

Figure 3 : le ficher et les proopriétés de : script\_generation………………………………….. 11

Figure 4 : Illustration Graphique de travail de (script\_generation)………………………….12

Figure 5 : Les ficher de LDRA WorkArea et LDRA ToolSuite…………………………………….12

Figure 6 : fichiers de configuration de projet pour LDRA…………………………………………13

Figure 7 : Les fichiers de TCF………………………………………………………………………………….14

Figure 8 : dossier de code source contient les fichiers .c……………………………………..….14

Figure 9 : Rapport d'analyse de la couverture du code………………………………………....15

Figure 10 : Rapport de régression d'unité LDRA…………………………………………………….…16

Figure 11 : Illustration Graphique de travail de l’application…………………………………..16

Figure 12 : sélection de dossier de rapportr\_HTML\_couverture…………………………….17

Figure 13 : Excel qui contient les résultats des fonctions testées avec leur …….……..17

taux de couverture

Figure 14 : Illustration Graphique de travail de l’application…………………………….…....19

Figure 15 : sélection de dossier de code source……………………………………………….……20

Figure 16 : exemple de tableau de fichier csv………………………………………………..……...21

Figure 17 : sélection de dossier LDRA-WorkArea…………………………………………………...23

Figure 18 : sélection de dossier PTU\_TCF………………………………………………………..…....24

Figure 19 : Les étapes de conversion un code python.py en un exécutable .exe….….25

Figure 20 : L’application final réalisé………………………………………………………………….......25

**CHAPITRE 1 : Présentation l’entreprise ACTIA Engineering Services**

**I. Introduction générale de l’entreprise :**

**ACTIA** est une entreprise française fondée en 1986, basée à Toulouse, spécialisée dans

le développement de solutions électroniques et numériques pour divers secteurs industriels.

Depuis sa création, **ACTIA Engineering Services** a connu une croissance significative

et s’est imposée comme un acteur clé dans ses domaines de compétence grâce

à son innovation constante et son engagement envers la qualité et le développement durable.

ACTIA est un modèle de réussite pour les entreprises technologiques, alliant savoir-faire,

vision stratégique, et un engagement profond envers l’excellence. En forgeant des partenariats

Solides, en développant des technologies innovantes, et en recevant des distinctions

prestigieuses, ACTIA prouve que le succès durable réside dans la capacité à allier excellence

technique et vision d’avenir.

ACTIA Engineering Services se positionne comme un acteur majeur et

influent dans le domaine des technologies électroniques et des systèmes embarqués, avec

une histoire marquée par l’innovation, l’excellence technique et un engagement

indéfectible envers la qualité.

****

**II. Les Compétences et Expertises De ACTIA Engineering Services :**

Actia est une entreprise de haute technologie reconnue pour son expertise dans le

développement et l’optimisation de solutions technologiques avancées. Avec une approche

innovante et une grande maîtrise des différentes disciplines de l'ingénierie . Actia se

positionne comme un acteur majeur dans plusieurs domaines clés de l'industrie. Découvrez

ci-dessous les principales compétences et activités de l'entreprise :

**II-1. Développement de Logiciels Embarqués :**

Le développement de logiciels embarqués est au cœur des activités d’Actia, où l’entreprise

met en œuvre des solutions sophistiquées pour des systèmes embarqués dans divers

secteurs d’application.

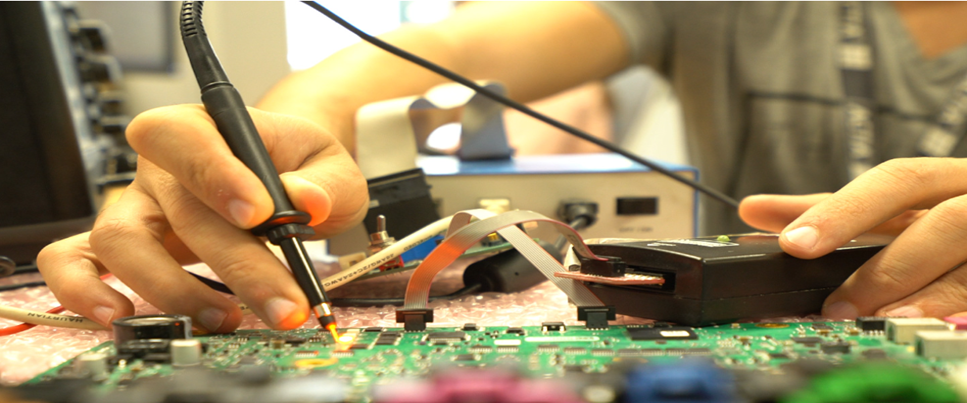
**\*- Conception de Systèmes de Contrôle Embarqués :**

Actia développe des logiciels qui gèrent le fonctionnement des systèmes électroniques

intégrés dans les véhicules, les équipements médicaux, et autres dispositifs technologiques.

Ces logiciels sont conçus pour être robustes, fiables et capables de fonctionner dans des

environnements contraints.



**II-2. Développement Web et Mobile :**

Actia développe des solutions web et mobiles innovantes pour répondre aux besoins des

entreprises et des utilisateurs finaux, avec une attention particulière portée à l’expérience

utilisateur.

**\*/** **Conception de Sites Web Responsives :**

Actia crée des sites web qui s’adaptent à tous les types d’appareils (ordinateurs, tablettes,

smartphones), offrant une interface utilisateur intuitive et un accès facile aux informations.

**\*/ Développement d’Applications Mobiles :**

L’entreprise développe des applications pour les plateformes iOS et Android, en mettant

l'accent sur l'ergonomie, la performance et la sécurité.

**II-3. Développement Électronique :**

Actia combine ingénierie mécanique et électronique pour concevoir des produits de

haute technologie et des systèmes intégrés.

**\*/ Développement de Circuits Électroniques :**

L'entreprise conçoit des circuits imprimés et des systèmes électroniques, en intégrant des

composants modernes pour répondre aux exigences de performance et de fiabilité.

**II-4. Réalisation de Tests et Validations Produits :**

Actia se consacre à la réalisation de tests rigoureux pour garantir la qualité et la conformité des produits.

**A / Tests Fonctionnels :**

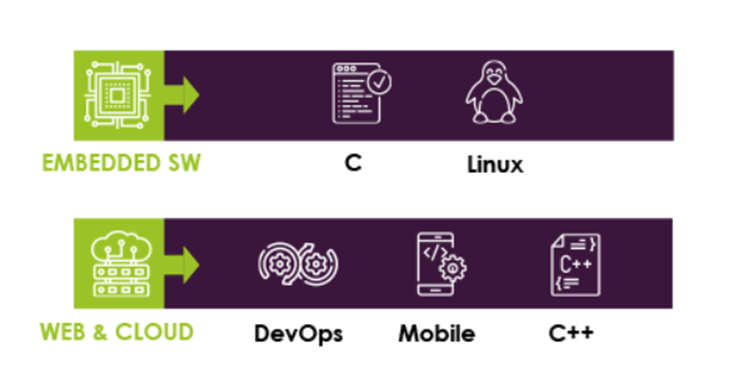
L’entreprise effectue des tests pour vérifier que les produits fonctionnent comme prévu dans

des conditions normales et extrêmes.

**B / Tests de Performance :**

Actia évalue la performance des systèmes et pour s'assurer qu'ils répondent

aux critères de performance et aux exigences des clients.



**\*/ Les valeurs de ACTIA Engineering Services :**



**III. Introduction Générale sur le Domaine du Test Unitaire :**

**Le test unitaire :** Est une pratique de test qui consiste à tester individuellement des unités de

code ’des fonctions’, pour vérifier leur bon fonctionnement.

Chez **ACTIA Engineering Services**, cette pratique est fondamentale pour garantir la qualité

et la fiabilité des codes pour des systèmes embarqués, particulièrement

dans différentes secteurs .

Le test unitaire assure la livraison de logiciels de haute qualité, fiables

et performants, répondant aux exigences élevées de l'industrie automobile et des transports.



**IV.** **Les outils utiliser en Test Unitaire :**

**\*/ LDRA :**

****

**Figure 1:** image se software LDRA

**\*/ Introduction à LDRA :**

LDRA (Liverpool Data Research Associates) est une suite d'outils de test et d'analyse de code,

utilisée principalement pour assurer la qualité et la conformité du logiciel dans des domaines

critiques.

LDRA joue un rôle crucial dans la vérification et la validation des logiciels , en fournissant

des outils avancés pour l'analyse, le test unitaire, et la conformité aux normes industrielles.

Les différentes versions et éditions des outils LDRA permettent aux développeurs de s'assurer

que leurs logiciels sont sûrs, fiables et conformes aux exigences réglementaires.

**\*/ LDRA pour les Tests Unitaires**

LDRA facilite les tests unitaires via :

Génération Automatique de Cas de Test : Crée automatiquement des cas de test.

Exécution et Validation des Tests : Exécute des tests avec des résultats détaillés.

Rapports de Test : Fournit des rapports détaillés sur les résultats des tests et la couverture de

code.

**\*/ Versions Utilisées de LDRA pour les Tests Unitaires**

**LDRA la Version utilisée 10.0.3 :**

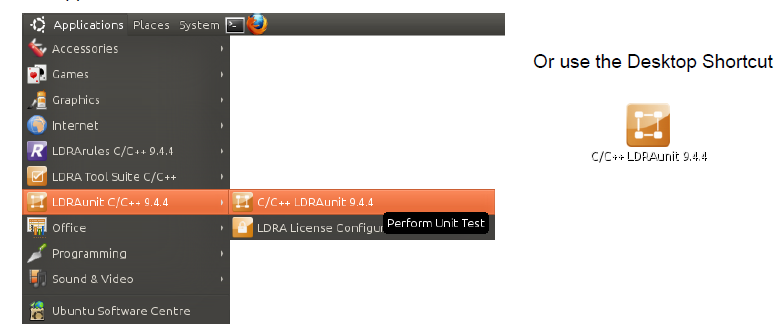
La version 10.0.3 est une des versions récentes qui inclut des améliorations en termes de

performance, de compatibilité et de fonctionnalité par rapport aux versions précédentes.

Elle offre des fonctionnalités avancées et des améliorations par rapport aux versions

précédentes .

**\*/ LDRA propose plusieurs outils pour répondre aux besoins :**

****

**Figure 2 :** Sélection de l’applications LDRAunit

**\*- LDRAunit :** Outil dédié aux tests unitaires, fournissant une solution complète pour la

création, l'exécution et l'analyse des tests unitaires.

**\*- ldraconunit :**, conçue pour faciliter l'intégration continue et l'automatisation

des tests unitaires. Cet exécutable permet de lancer les tests unitaires dans des environnements.

**Chapitre 2 : Cadre du projet**

**\*/ Titre du projet :**

Ajustement du script d’automatisation de test unitaire

**I. Introduction :**

Les tests unitaires jouent un rôle crucial pour vérifier la fonctionnalité correcte de chaque

composant du logiciel. Pour simplifier et automatiser ce processus, des scripts spécialisés sont

souvent utilisés. Pour gérer efficacement ces tests, des outils d'automatisation sont souvent

utilisés pour exécuter les tests, collecter les résultats, et générer des rapports. Dans ce projet,

le but était de fusionner deux scripts distincts en un seul script Python afin d'améliorer

l’efficacité et l'intégrité du processus de test.

**\*/ Le processus de test était précédemment divisé en deux scripts :**

**1.Script .bat :** Automatise l'exécution des tests unitaires en appelant test.exe avec

chaque fichier de test .tcf et copie les résultats dans un répertoire de rapports.

**2.Script Python .py :** Analyse les rapports HTML  générés par les tests .

**\*/** Le projet de stage a eu pour but de fusionner ces deux scripts en un

seul script Python afin d’améliorer l’automatisation et l’efficacité du processus de test.

**II. Objectifs du Projet :**

**1.Automatiser l'Exécution des Tests :** Remplacer le script linux et le script .Py par un script

python capable de gérer automatiquement l'exécution des tests.

**2.Fusionner les Fonctionnalités :** Combiner les fonctionnalités des deux scripts en un

**3.Assurer la Robustesse :** Ajouter des mécanismes de gestion des erreurs et des

exceptions pour rendre le processus plus fiable.

**4. Documenter le Code :** Fournir une documentation complète pour aider à la

maintenance et à l’évolution future du script.

**III. Les étapes du Projet :**

1. **Analyse des Scripts Existants :** Examiner les deux scripts pour comprendre leurs

fonctions, identifier leurs faiblesses, et repérer les opportunités d’amélioration.

1. **Conception du Nouveau Script :** Définir l’architecture du nouveau script Python,

en intégrant les fonctionnalités des deux scripts et en planifiant les optimisations nécessaires.

1. **Développement du Script :** Écrire le code du script en implémentant

les fonctionnalités nécessaires, en ajoutant la gestion des erreurs, et en optimisant

les performances.

1. **Tests et Validation :** Vérifier que le script fonctionne correctement à travers divers

tests, en mesurant ses performances et en s’assurant qu’il répond aux spécifications.

**IV. Contexte du Projet :**

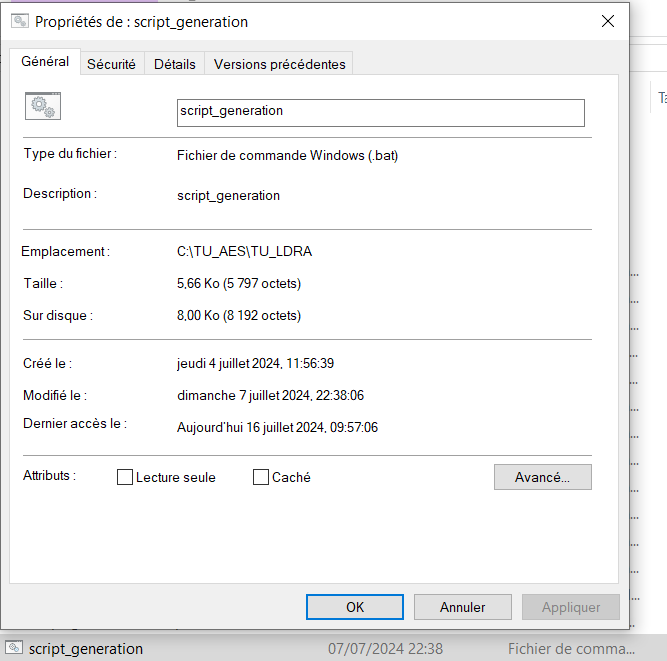
**IV. 1-** **Analyse du Script .bat(script\_generation) :**

C’est un fichier de commande Windows (.bat)

Le code fourni est un script batch (.bat), utilisé principalement dans les systèmes

d'exploitations Windows pour exécuter des commandes dans l'interpréteur de commandes

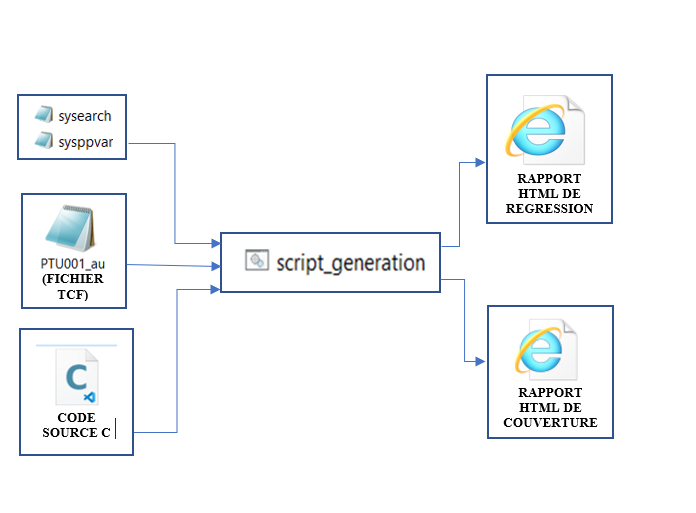
de Windows.

 **Figure 3 :** le ficher et les proopriétés de : script\_generation

**Ce script.bat :** est utilisé pour automatiser l'exécution de tests LDRA (un outil de test et

d'analyse pour le développement de logiciels embarqués) et pour générer des rapports de

**couverture dynamique** et de **régression.**



**Figure 3 :** Illustration Graphique de travail de (script\_generation)

**IV.1.1-** **Préparation de l'environnement :**

**\*/ Création des répertoires :**

mkdir %Rapports\_HTML\_Couverture%

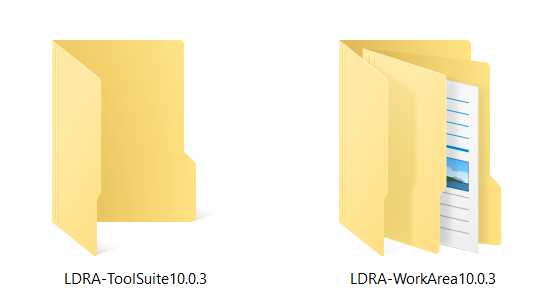
mkdir %Rapports\_HTML\_Regression\_Interphone%

Crée les répertoires pour les rapports de couverture et de régression.

**\*/ Définition des variables globales :**

INSTALL\_LDRA=C:\LDRA-ToolSuite10.0.3 : Chemin d'installation de LDRA.

WORKAREA\_LDRA=C:\LDRA-WorkArea10.0.3 : Chemin de zone de travail LDRA



**Figure 5:** Les ficher de LDRA WorkArea et LDRA ToolSuite

**\*/**  **LDRA WorkArea :**

Le LDRA Work Area (Zone de Travail LDRA) est un espace de travail virtuel dans le

LDRA Tool Suite où les utilisateurs peuvent configurer, gérer et exécuter leurs projets de test.

Il représente l'environnement dans lequel toutes les activités de test et d'analyse se déroulent.

**\*/**  **LDRA Tool Suite :**

LDRA Tool Suite est un ensemble de logiciels développés par LDRA , conçu pour aider au

développement, à la vérification et à l'analyse de logiciels embarqués. Cette suite d'outils est

utilisée principalement dans des environnements de développement de logiciels critiques où la

qualité et la conformité aux normes sont essentielles.

**IV.1.2-** **Exécution des tests TBrun :**

call:executeTbrun audio\_alvoy\_conducteur\_bi.c audio\_alvoy\_conducteur\_bi

🡪 Appelle la fonction executeTbrun pour exécuter les tests sur les fichiers spécifiés.

**IV.1.3- /** **Exécution de LDRA :**

Lance LDRA avec les fichiers et paramètres spécifiés :

start /Wait %INSTALL\_LDRA%/conunit.exe %source\_path\_base%\%~1

/SYSEARCH="%path\_cfg\_ldra\_sysearch\_use%"

/SYSPPVAR="%path\_cfg\_ldra\_sysppvar\_use%"



**Figure 5 :** fichiers de configuration de projet pour LDRA

**\*/ sysearch** : Liste des chemins dossiers où chercher des fichiers nécessaires pour la compilation ou l'analyse dans les systèmes de développement.

**\*/ sysppvar** : Contient des variables qui définirent des options ou paramètres pour la configuration du système.

**\*/ Les TCF (Test Case Files) :**

Les Fichiers de Cas de Test sont utilisés pour décrire les scénarios de test.



**Figure 7 :** Les fichiers de TCF

**IV.1.4-** **Exécution des fichiers TCF :**

\*Exécution des fichiers TCF :

Cela permet de tester tous les fichiers .tcf dans un répertoire spécifique, d'exécuter des tests

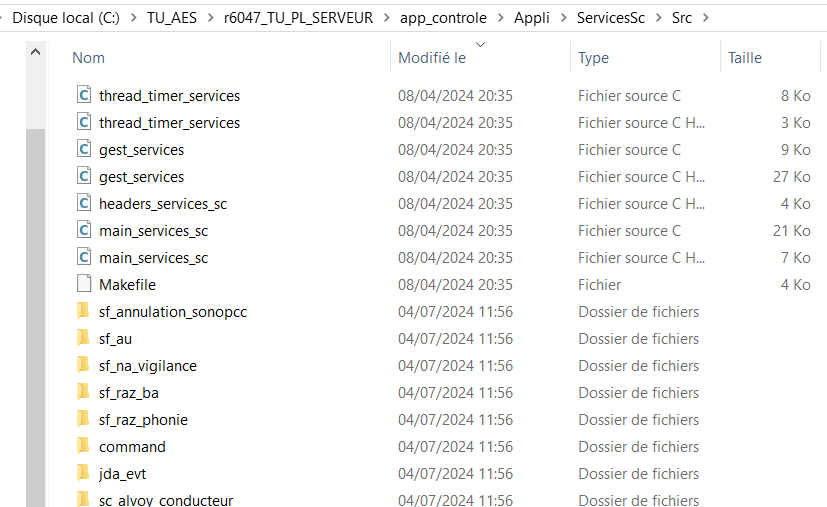
\*automatisés avec LDRA :

Parcourt tous les fichiers TCF dans le répertoire et les exécute avec LDRA.

LDRA permet de générer automatiquement des TCF en se basant sur les exigences définies

et **le code source** disponible. Cette génération automatique assure que les cas de test couvrent

une large gamme de scénarios incluant **les chemins critiques** du code .



**Figure 8 :** dossier de code source contient les fichiers .c

**IV.1.5-** **Extraction et copie des rapports :**

Parcourt les répertoires de résultats et copie les fichiers HTML de couverture et de

Régression vers les répertoires de sortie.

**\*/ Le rapport de couverture dynamique :**

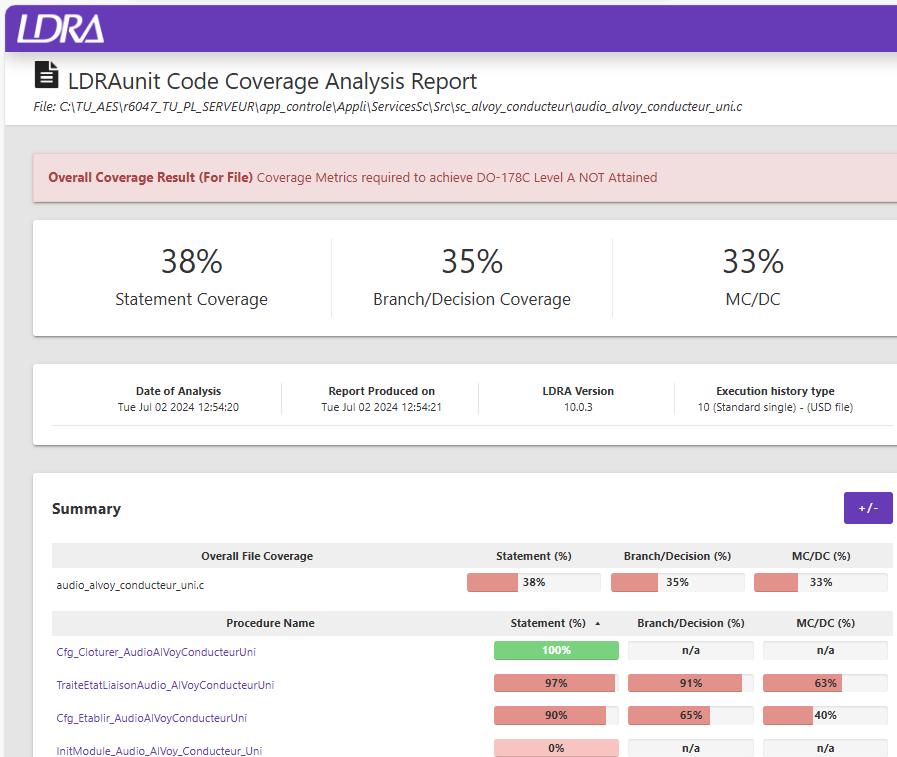
Est un document généré par des outils de test (LDRA) pour montrer dans quelle mesure le

code source d'un programme a été exercé lors des tests dynamiques. La couverture dynamique

est une mesure de l'étendue à laquelle le code source est exécuté lors des tests. Le rapport

HTML fournit une vue détaillée de la couverture du code, y compris des informations sur les

parties du code qui ont été testées et celles qui ne l'ont pas été.

****

**Figure 9:** Rapport d'analyse de la couverture du code

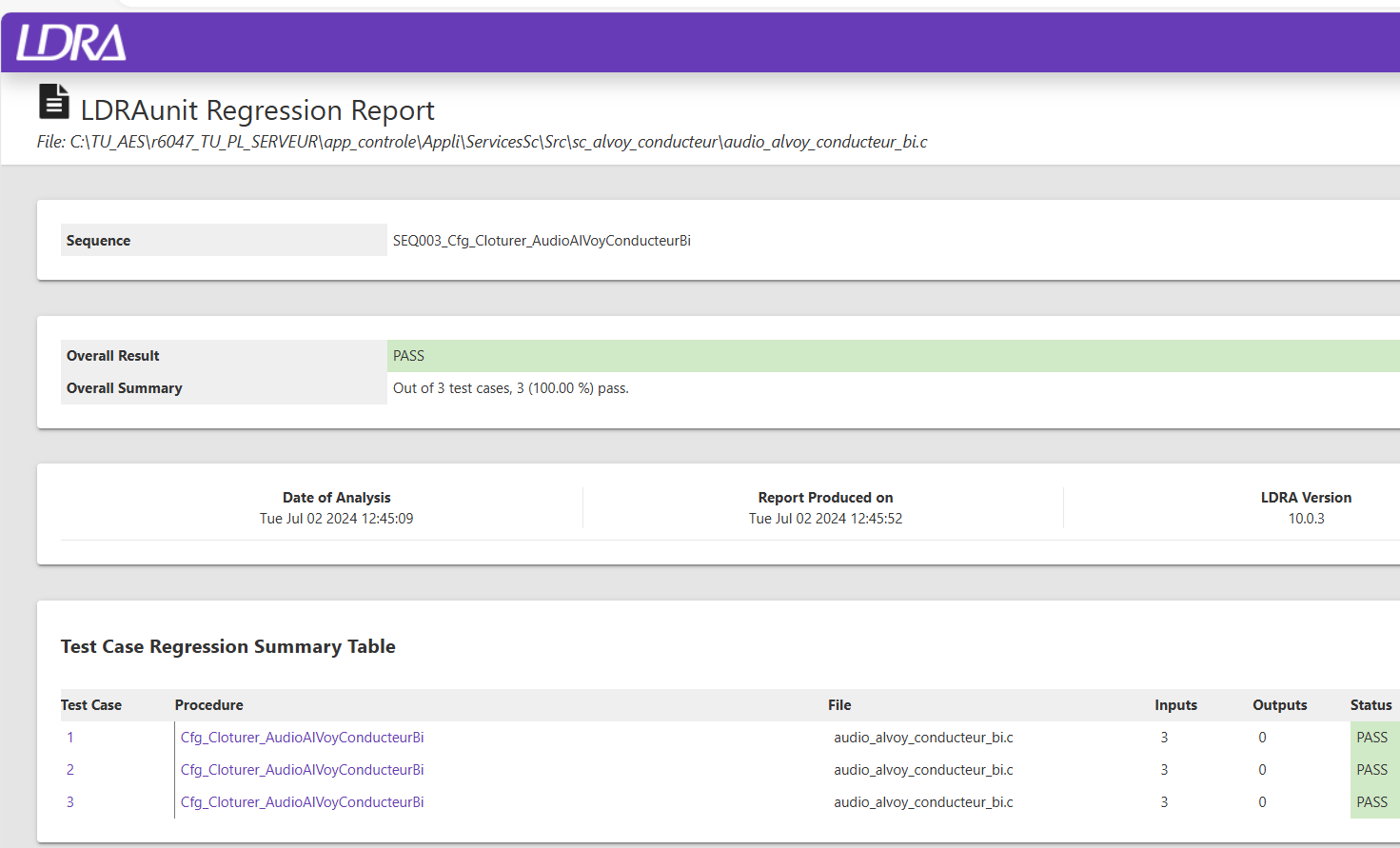
**\*/ Le rapport de régression** :

Est un document généré par des outils de test (LDRA) pour montrer les résultats des tests

de régression effectués sur le logiciel. Les tests de régression vérifient que les modifications

apportées au code n'ont pas introduit de nouveaux erreurs et que les fonctionnalités existantes

fonctionnent toujours comme prévu.

****

**Figure 10 :**  Rapport de régression d'unité LDRA

**IV.2-** **Analyse du Script Python (script\_synthese) :**

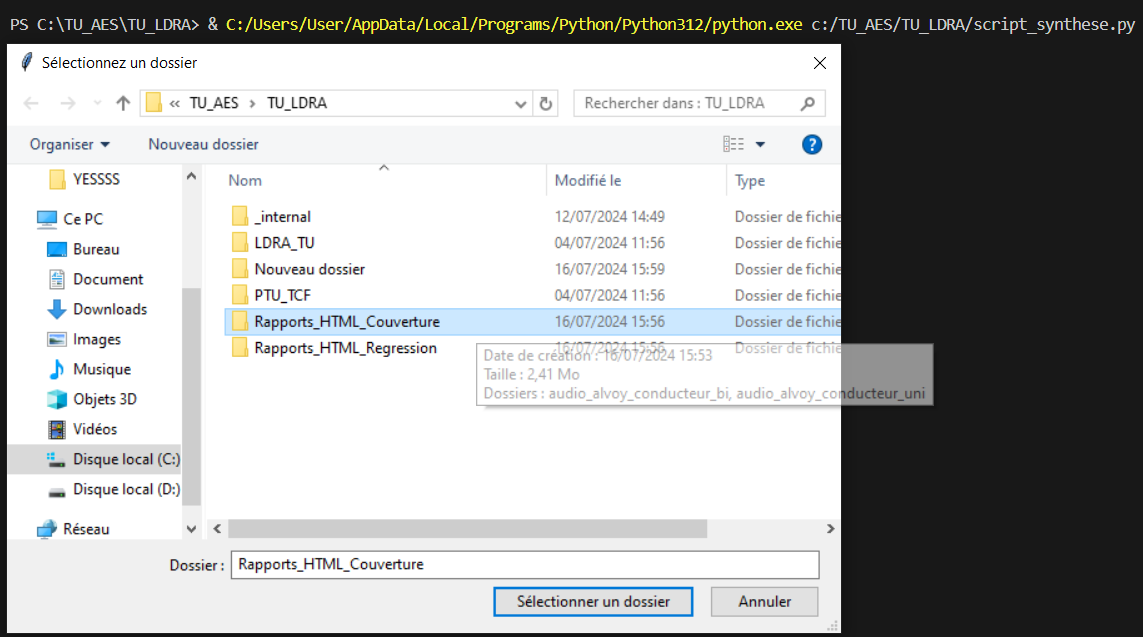
Ce script Python a pour objectif d'extraire des données à partir des fichiers HTML situés

dans le dossier Rapports\_HTML\_Couverture et de les présenter dans un fichier Excel.

****

**Figure 11 :** Illustration Graphique de travail de l’application

**\*- Selectionner manuellement ce dossier :**



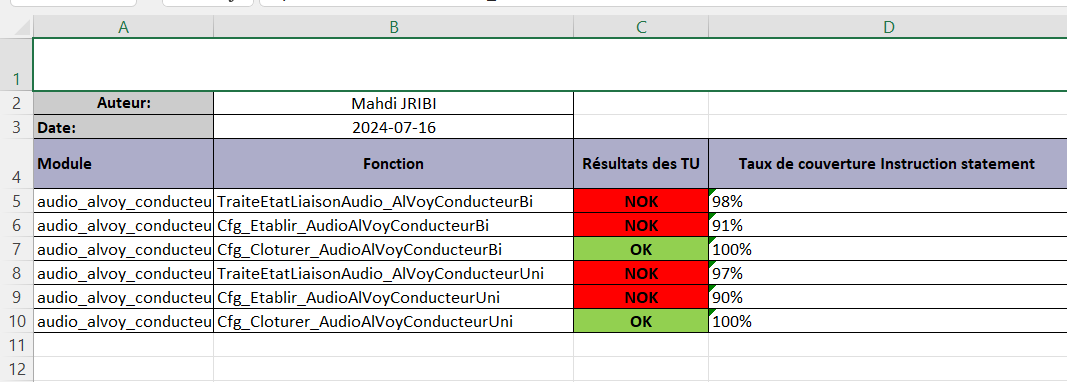
**Figure 12 :** sélection de dossier de rapportr\_HTML\_couverture

**Et puis sélectionnez ou tu veux enregistre le ficher Excel (RTU) :**

**IV.2.1- Le résultat final :**

Un fichier Excel qui contient les résultats des fonctions testées avec leur taux de couverture





**Figure 13 :** Excel qui contient les résultats des fonctions testées avec leur taux de couverture

**\*/ Taux de Couverture Instruction :**

Le taux de couverture d'instruction mesure le pourcentage des lignes de code qui ont été

exécutées pendant les tests. Il est une mesure basique mais importante de la qualité des tests,

indiquant si chaque ligne du code source a été exécutée par les tests.

**V. Problématique :**

La séparation des processus de test entre un script .bat et un script .py entraîne

des problèmes.

**•Inefficacités dans le flux de travail :** Nécessité d’exécuter manuellement deux scripts

séparés, introduisant des risques d'erreurs humaines et des inefficacités.

**•Limitations de performance :** Exécution séquentielle des tests, augmentant le temps

requis pour traiter une suite de tests volumineuse.

**•Gestion des Erreurs :** Le script .bat ne gérait pas les erreurs qui pouvaient survenir

lors de l'exécution des tests. Le script Python supposait que les rapports HTML étaient

toujours corrects.

**•Complexité de Maintenance :** Avoir deux scripts distincts compliquait

la maintenance et l’évolution des outils de test.

**Chapitre 3 : Exécution du Projet**

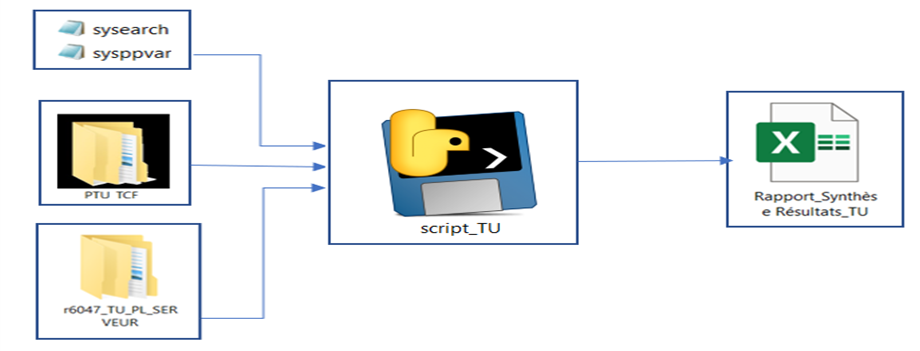
**I. Conception du Nouveau Script python :**

Le nouveau script Python remplace le script .bat (script\_generation ) et le script.py

(script\_synthese) et automatisant l’exécution des tests et l’analyse des rapports en un seul

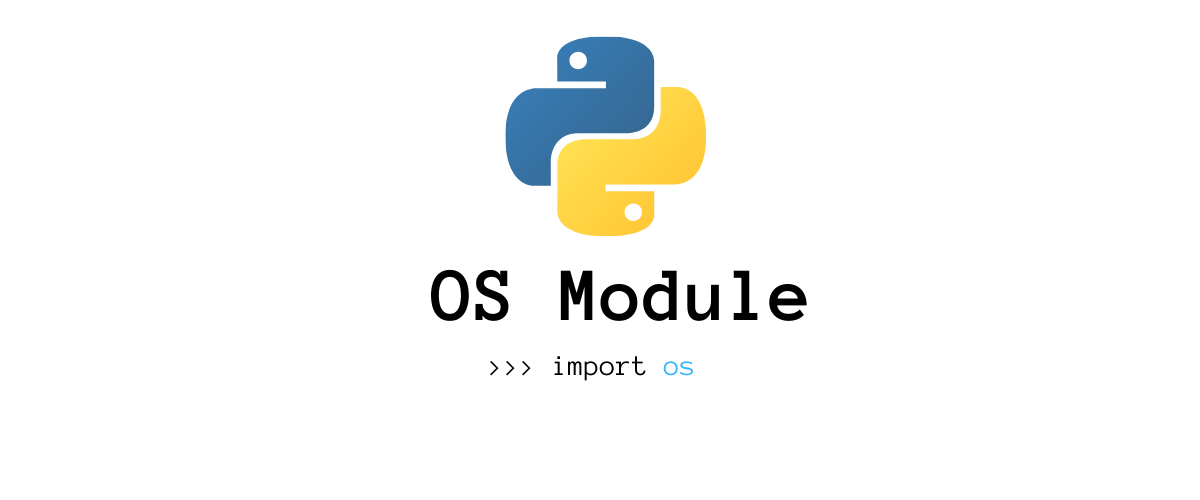
processus . La centralisation des processus simplifie le flux de travail et réduit les risques

d’erreurs manuelles, rendant le processus de test plus efficace et plus fiable.



**Figure 14 :** Illustration Graphique de travail de l’application

** II. explication des importations utilisée :**

**\*- import datetime :** Ce module fournit des classes pour manipuler les dates et les heures.

**\*- import os :** fonction système d'exploitation avec python

****

**\*- import openpyxl :** manipuler les fichier Excel

****

**\*- import bs4 :** conçu pour travailler avec des documents HTML

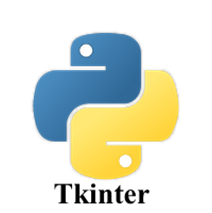
****

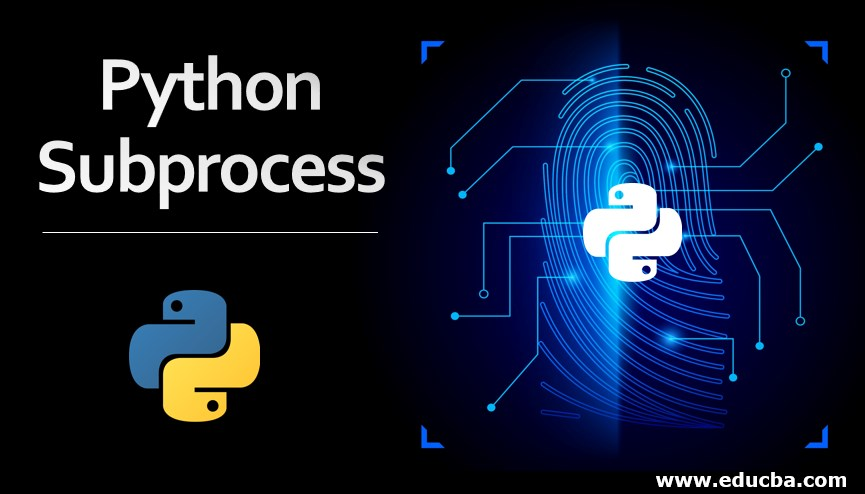
**\*- import shutil :** utilisée pour des opérations de haut niveau sur les fichiers et les collections

****de fichiers

**\*- import pandas:** être très utile pour extraire et organiser les données des fichiers

HTML avant de les écrire dans un fichier E

**\*- import Tkinter :** Pour créer des interfaces graphiques permettant à l'utilisateur de sélectionner un répertoire.

****

**\*/ Import subprocess :** cela pourrait être utilisé pour exécuter LDRA

**III. Transformer le code .bat en code Python :**

****

12

**III.2- Explication des fonctions :**

17

**\*/ Fonction select\_directory () :**

Cette fonction adopter le dossier PTU\_TCF , chercher et lister tous les nom des fichiers

présents dans ce dossier. Les noms de fichiers .c sont ajoutés à une liste

**\*/ Fonction find\_files (directory, filenames) :**

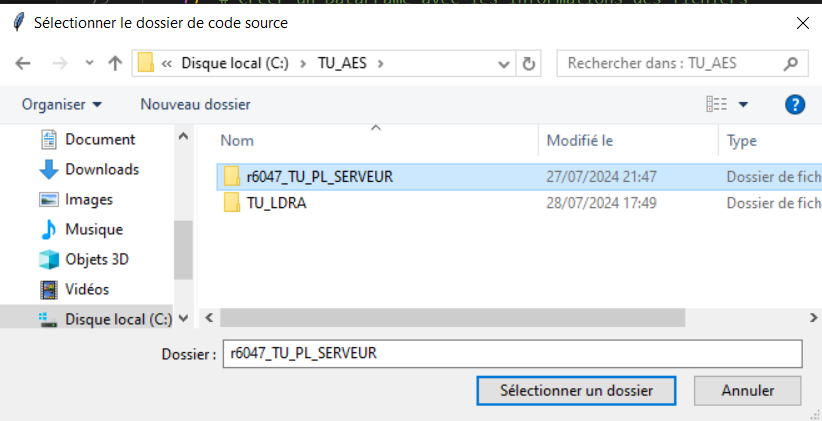
Elle parcourt le répertoire pour chercher des fichiers .c. Pour chaque fichier trouvé,

elle enregistre son nom dans une liste et leur chemin dans une autre liste .

**\*/ Fonction Sélectionner\_le\_dossier :**

La fonction fait La gestion du répertoire de **code source**, elle inclut la fonction **find\_files ()** et faire la vérification avec les noms de la liste de fichier de **ptu\_tcf** de la fonction **select\_directory ().**

**select\_directory() .**

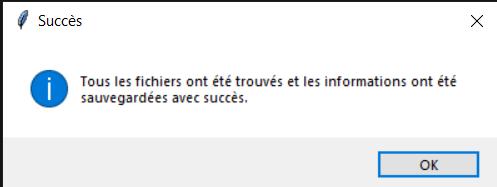


**Figure 15 :** sélection de dossier de code source

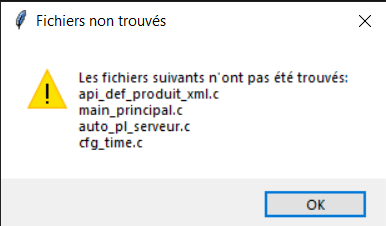
La fonction affiche un message d'avertissement si certains fichiers spécifiés ne sont pas

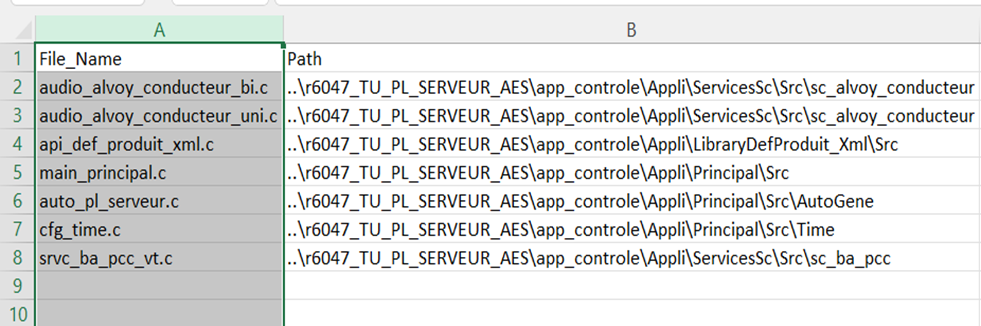
trouvés **ou** un message de succès si tous les fichiers ont été trouvés et les informations

correctement sauvegardées dans un fichier csv .

****

**Ou**

****



**Figure 16 :** exemple de tableau de fichier csv

**\*/ La fonction : lire\_et\_afficher\_csv () :**

Est conçue pour lire un fichier CSV (Excel) et extraire les file-Name (les ficher .c) et les path

, et afficher ces informations dans 2 liste .

**\*/ la fonction : execute\_tbrun(file\_name, file\_path):**

La fonction execute\_tbrun est conçue pour exécuter TBrun, un outil de tests unitaires et

d'intégration faisant partie de la suite d'outils LDRA, sur un fichier source spécifique (.c)

et un fichier de séquence associé. TBrun automatise la génération et la gestion des tests

unitaires .



**Fonctionnalités de la fonction execute\_tbrun :**

\* Suppression des répertoires existants pour les rapports

\* Création de nouveaux répertoires pour les rapports

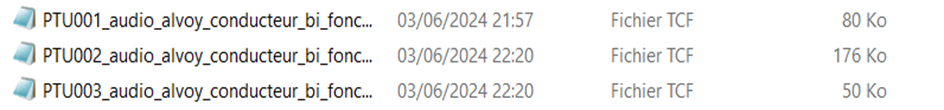
\* Exécution de l'outil LDRA

\* Gestion des erreurs : En cas d'échec, une erreur est imprimée

**\*/ la fonction : execute\_tcf\_files(file\_name):**

- Exécutez tous les fichiers TCF pour un fichier de séquence donné.

- Recherche et exécute tous les fichiers .tcf du répertoire avec LDRA .



- Appelle loop\_last\_token pour extraire le nom de la fonction.

**\*/ la fonction : copy\_reports(file\_name) :**

Cette fonction copie les rapports de régression et de couverture dynamique dans les

répertoires correspondants.

Recherche des rapports : La fonction recherche les répertoires contenant les rapports.

Copie des fichiers : Les fichiers .htm et .dyn.htm sont copiés dans les répertoires appropriés .

**IV. Amélioration du code (script\_synthese) :**

L'objectif est de rendre le code plus dynamique en sélectionnant automatiquement

le fichier de couverture et en enregistrant le fichier Excel de manière automatique

dans le répertoire de travail. Au lieu de définir le chemin de couverture manuellement

, le nouveau code pourrait utiliser un sélecteur de fichiers pour choisir le fichier

De couverture HTML automatiquement. Le code devra ensuite enregistrer le fichier Excel

****d’une façon dans le répertoire où le travail est effectué.

Il est nécessaire aussi d’améliorer la qualité et la robustesse de ce code

**IV.1- Explication des fonctions :**

**\*- la fonction :access\_folder\_in\_path(base\_path, folder\_name)**

Accéder à un dossier spécifique dans un chemin de base donné, conçue notamment pour

accéder à des répertoires qui contenir des fichiers Excel .

**\*- La fonction : extract\_tables\_to\_excel (folder\_path, output\_file):**

Cette fonction est conçu pour extraire des données à partir de fichiers HTML

****et les enregistrer dans un fichier **Excel** avec un formatage spécifique.

**\*- la fonction : process\_excel\_file(file\_path):**

La fonction est conçue pour traiter et nettoyer un fichier Excel

plusieurs étapes, en utilisant différentes fonctions utilitaires pour adapter les données

et préparer le fichier pour une utilisation ultérieure. Voici un aperçu détaillé de chaque étape

de la fonction en expliquant l’utilisation ces fonctions :

* 1. **delete\_empty(ws) :**Supprime les lignes vides de la feuille de calcul.
  2. **add\_percent(ws) :**Ajoute un format de pourcentage ou calcule des valeurs en pourcentage dans la feuille calcul.
  3. **updating(ws) :**Met à jour la feuille de calcul.

**V. les modifications réalisées de code final :**

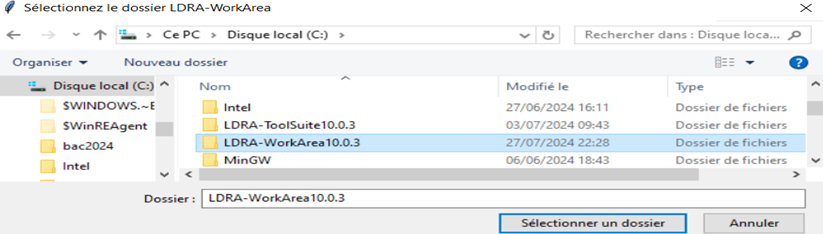
**\*/ En ajoute la fonction select\_directory(title):**

Pour simplifier l'utilisation du script et améliorer l'interaction avec l'utilisateur,

nous avons intégré une fonctionnalité permettant la sélection des répertoires

LDRA-WorkArea et LDRA-ToolSuite avec une interface graphique.

**\*/ Exemple de affichage et de sélection :**



**Figure 17 :** sélection de dossier LDRA-WorkArea

Aussi en ajoute à la fonction select\_directory ()de sélectionner le fichier "ptu\_tcf "



**Figure 18 :** sélection de dossier PTU\_TCF

La modification vise à améliorer ce programme en permettant à l'utilisateur de sélectionner un

dossier PTU\_TCF .

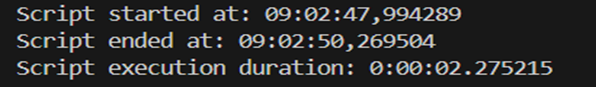
**\*/ Affichage des Chemins :**

Afficher les chemins définis pour vérifier que les répertoires sont correctement

configurés. Cela aide à vérifier que les chemins sont corrects avant d'exécuter des opérations

de fichier ou de dossier.

**V.1- Affichage de l'Heure de Début et de Fin du Script :**



**\*/ but :**

\*Analyse de Performance .

\*Mesurer la durée d'exécution est essentiel pour :

\*Comparer les performances entre différentes versions du script pour détecter

des améliorations ou des régressions.

**V.2- Gestion des Erreurs :**

Introduction de la gestion des erreurs pour assurer la robustesse

**\*/ Exemple :**

•Bloc **try-except :**

•Les instructions conditionnelles **if et if not :**

Les structures de Gestion des Erreurs sont des outils puissants pour la prise de

décision dans le code Python et pour créer des programmes plus robustes, flexibles et faciles

à maintenir. Ces structures améliore la logique et le flux d'exécution de vos programmes tout

en assurant une meilleure gestion des conditions spécifiques et des scénarios d'erreur.

**Exemple d’affichage indiquer de n’est pas trouver le fichier csv à l'emplacement spécifié :**

****

**V.3- Documentation et Maintenance :**

Le script est bien documenté, avec des commentaires explicatifs et une documentation

détaillée. Une bonne documentation facilite la maintenance et l’évolution future du script,

permettant à d'autres développeurs de comprendre et de modifier le code selon les besoins.

**VI. convertir le code python .py en un exécutable .exe :**

**\*/ Les étapes de conversion :**

1. mètre le code.py python dans un dossier « code » dans le Desktop

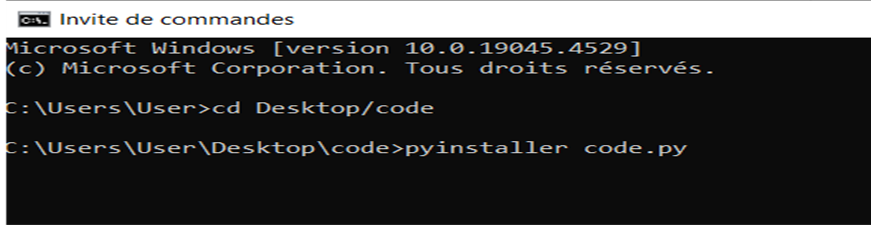
2. on tape sur la recherche CMD

3. ouvrire CMD (invite de commandes)

4. taper > pip install pyinstaller

5. taper > cd Desktop / code

6. taper > cd pyinstaller code.py



**Figure 19 :** Les étapes de conversion une application python .py en un exécutable .exe

**\*/ Voisi l’application final réalisé :**

****

**Figure 20 :** L’application final réalisé

**VII. Conclusion :**

Le projet a été couronné de succès. La fusion des deux scripts a permis d’atteindre les

objectifs initiaux et de répondre efficacement aux problématiques identifiées.

En conclusion, le projet de fusion des scripts permise de créer une solution

plus intégrée et performante pour l'exécution des tests et l'analyse des résultats.

Aussi fourni une base solide pour une maintenance future et des améliorations

ultérieures.

Le succès de ce projet démontre l'importance de l’automatisation dans le

processus de développement logiciel et la valeur de la gestion efficace des tests pour garantir

la qualité du produit final.

L'expérience acquise au cours de ce projet a été extrêmement précieuse, en offrant des

perspectives sur l'amélioration des processus de test, la gestion des erreurs, et l'optimisation

des performances. Ces compétences seront utiles pour des projets futurs et pour le

développement de solutions d'automatisation plus sophistiquées.

13

**\*/**