# Sommaire :

Sommaire : 1

Intro : 1

Détail du travail effectué : 1

Git hub 3

Labview 4

Notes lab view 4

Application android 4

Conclusion : 4

Annexes 4

Glossiare 4

Github 4

LabView 7

Android studio 8

Bibliographie 9

Github 9

Labview 9

android studio 9

Résumé 9

# Introduction

# I Détail du travail effectué

Dans cette section nous allons détailler le déroulement du travail effectué durant le 2eme semestre

Tableau d’attribution de lettres aux étudiants

|  |  |
| --- | --- |
| RAMOS GONCALVES Wilson | W |
| IM-SAROEUN Paul | P |
| NICOLAS Jules | J |
| BELLANGER Liam | L |

Tuteur : M. MOHAMMED

Tableau des réunions effectuées

|  |  |  |  |
| --- | --- | --- | --- |
| Réunion | Thème | Contributeurs | Dates |
| 1 | Répartition des tâches | W,P | Jeudi 09/12 |
| 2 | Recherches + début d’application | W,P | Dimanche 12/12 |
| 3 | Mise en commun | W,P,J | Lundi 03/01 |

**Rapport de réunion Jeudi 09/12:**

Discussion sur les éléments qu’on va mettre dans le rapport :

Vidéo du fonctionnement de l'orthèse

Explication du fonctionnement du code labview

Explication de l'implémentation java labview

Creation de l’app

Répartition du travail:(pour l’instant wilson et paul)

Paul

 Développement de l'application android

Explication du fonctionnement

Wilson

Recherches sur l'implémentation app Labview

Recherches sur mode de communication

 Rapport de réunion Jeudi 12/12:

-Paul

    Début de dev de l’application

    Création d’un repository github

-Wilson

rédaction rapport + recherches

Création d’un repository github

 Rapport de réunion Jeudi 03/01:

-Wilson

    Mise en place du repository github

    Labview

-Paul

    dev application

-Jules

    Début de rédaction rapport

Rapport de réunion Jeudi 14/01:

-Wilson

    Dev application

Redaction rapport

-Paul

     Dev application

    Rédaction rapport

Wilson

## 1.1 Git hub

Fonctionement de repositories git

Comment on est 3 on avait besoin d’un outil nous permettant de travailler ensemble tout en synchronisant notre travail.

Nous avons donc choisi git je vais donc expliquer son fonctionnement :

 I Présentation

II répartition

Présentation

Git est un logiciel de versioning qui permet de conserver un historique des modifications effectués sur un projet afin de pouvoir identifier les changements effectués et de revenir a une ancienne version en cas de problème

Il nous permet donc :

- D’éviter les conflits entre la version

- Informer sur quoi chaque membre du groupe travail

- Permettre que chaque membre du groupe d’avoir une version actualisée du code

Fonctionnement

On a différentes commandes :

Git init : qui permet de démarrer un repo

## 1.2 Labview

Notes lab view

Crtl + T: side by side

Crtl + e: afficher front panel ou diagram

On a choisi le protocole tcp car il est plus fiable et dans le cas de notre application cela est essentiel

## 1.2 Application android

# II CONCLUSION

# III Annexes

# IV Glossiare

## 4.1 Github

**Access token**

Un token qui est utilisé à la place d’un mot de passe pour effectuer des opérations via HTTPS avec GIT dans un terminal ou une API

**Branche**

**Une branche est une version parallèle d'un dépôt. Elle est contenue dans le dépôt, mais n'affecte pas la branche primaire ou principale, ce qui nous permet de travailler librement sans perturber la version "courante". Lorsque on a apporté les modifications souhaitées, on peut fusionner la branche avec la branche principale pour publier les changements.**

**Checkout**

On utilise git checkout dans un terminal pour créer une nouvelle branche, changer la branche de travail actuelle pour une branche différente, ou même pour passer à une version différente d'un fichier provenant d'une branche différente avec git checkout [nom de la branche] [chemin du fichier]. La commande "checkout" met à jour tout ou une partie de l'arbre de travail avec un objet ou un blob de la base de données des objets, et met à jour l'index et le HEAD si l'arbre de travail entier pointe vers une nouvelle branche.

**Clone**

Un clone est une copie d'un dépôt sur notre ordinateur plutôt que sur le serveur de github, ou l'acte de faire cette copie. Lorsqu'on fait un clone, on peut éditer les fichiers dans son éditeur préféré et utiliser Git pour garder la trace de ses modifications sans avoir à être en ligne. Le dépôt qu'on a cloné est toujours connecté à la version distante de sorte qu'on peut push les modifications locales vers la version distante pour les synchroniser quand on est en ligne.

**Collaborateur**

Un collaborateur est une personne ayant un accès en lecture et en écriture à un dépôt et qui a été invitée à contribuer par le propriétaire du dépôt.

**Commit**

Un commit, ou "révision", est une modification individuelle d'un fichier (ou d'un ensemble de fichiers). Lorsqu'on réalise un commit pour sauvegarder son travail, Git crée un identifiant unique (alias le "SHA" ou "hash") qui permet de garder une trace des modifications spécifiques commises ainsi que de l'auteur et de la date. Les commits contiennent généralement un message de commit qui est une brève description des changements effectués.

**Git**

Git est un programme open source pour le suivi des changements dans les fichiers texte. Il a été écrit par l'auteur du système d'exploitation Linux et constitue la technologie de base sur laquelle est construite GitHub, l'interface sociale et celle de l'utilisateur.

**Pages GitHub**

Également appelé Pages. Un service d'hébergement de sites statiques conçu pour héberger vos pages personnelles, d'organisation ou de projet directement à partir d'un dépôt GitHub.

**Main**

La branche de développement par défaut. Chaque fois que vous créez un dépôt Git, une branche nommée main est créée, et devient la branche active. Dans la plupart des cas, elle contient le développement local, bien que cela soit purement par convention et ne soit pas nécessaire.

**Merge**

La fusion prend les changements d'une branche (dans le même dépôt ou dans une branche secondaire), et les applique à une autre. Cette opération s'effectue souvent sous la forme d'une "pull request" (que l'on peut considérer comme une demande de fusion), ou via la ligne de commande. Une fusion peut être effectuée par une demande de pull via l'interface web de GitHub.com s'il n'y a pas de changements conflictuels, ou peut toujours être effectuée via la ligne de commande.

**Merge conflict**

Une différence qui se produit entre les branches fusionnées. Les conflits de fusion se produisent lorsque des personnes apportent des modifications différentes à la même ligne d'un même fichier, ou lorsqu'une personne modifie un fichier et qu'une autre personne supprime le même fichier. Le conflit de fusion doit être résolu avant que vous puissiez fusionner les branches.

**Open source**

Les logiciels libres sont des logiciels qui peuvent être librement utilisés, modifiés et partagés (sous forme modifiée ou non) par quiconque. Aujourd'hui, le concept de " open source " est souvent étendu au-delà des logiciels, pour représenter une philosophie de collaboration dans laquelle les matériaux de travail sont mis à disposition en ligne pour que chacun puisse les utiliser, les modifier, en discuter et y contribuer.

**Pull**

Pull fait référence au moment où l'on récupère des modifications et où on les fusionne. Par exemple, si quelqu'un a modifié le fichier distant sur lequel on travaille tous les deux, on voudra transférer ces modifications dans la copie locale pour qu'elle soit à jour.

**Pull request**

Les demandes de Pull sont des propositions de modifications d'un référentiel soumises par un utilisateur et acceptées ou rejetées par les collaborateurs du référentiel. Comme les questions, les demandes de retrait ont chacune leur propre forum de discussion.

**Push**

Push signifie qu'on envoie les modifications apportées à un dépôt distant sur GitHub.com. Par exemple, si on modifie quelque chose localement, on peut pousser ces modifications pour que d'autres puissent y accéder.

**Repository**

Un dépôt est l'élément le plus fondamental de GitHub. Il est plus facile de l'imaginer comme le dossier d'un projet. Un dépôt contient tous les fichiers du projet (y compris la documentation), et stocke l'historique des révisions de chaque fichier. Les dépôts peuvent avoir plusieurs collaborateurs et peuvent être publics ou privés.

## 4.2 LabView

**Boucle While**

Structure en boucle qui répète une section de code jusqu'à ce qu'une condition ait lieu.

**Bouts de fils de liaison**

Fils de liaison tronqués qui apparaissent à côté de terminaux non câblés lorsque vous déplacez l'outil Bobine sur un VI ou un nœud de fonction.

**Connecteur**

Partie du VI ou du nœud de fonction qui contient les terminaux d'entrée et de sortie. Les données passent par un nœud et sortent d'un nœud par le biais d'un connecteur.

Zone dans le coin droit supérieur de la face-avant ou du diagramme qui affiche le modèle des terminaux du VI. Il définit les entrées et les sorties que vous pouvez câbler à un VI.

**Entier**

Tous les nombres entiers naturels, leur opposé, ou zéro.

**Exécution continue**

Mode d'exécution dans lequel un VI s'exécute de manière répétitive jusqu'à ce que l'opérateur l'arrête. Cliquez sur le bouton **Exécuter en continu** pour exécuter un VI de manière répétitive.

**Incrément de temps**

Intervalle de t à t + dt.

**Info-bulle**

Petites bannières de texte jaunes qui identifient le nom du terminal et facilitent l'identification des terminaux pour le câblage.

**LabVIEW**

Laboratory Virtual Instrument Engineering Workbench — LabVIEW est un langage de programmation graphique qui utilise des icônes au lieu de lignes de texte pour créer des applications.

**TCP/IP**

Transmission Control Protocol/Internet Protocol — Format standard de transmission de données en packets d'un ordinateur à un autre. Les deux composantes du TCP/IP sont le TCP, qui concerne la construction de paquets de données, et le IP, qui les achemine d'un ordinateur à un autre.

**UDP**

User Datagram Protocol

## 4.3 Android studio

**Android SDK**

Android SDK, ou « Software Developement Kit » du français Kit de développement logiciel), est une boîte à outil qui permet aux développeurs de tester leurs applications sur différentes versions d'Android, en les émulant. Le SDK regroupe les outils nécessaires à la connexion et à l'envoi de commandes ADB. Il est utile aux utilisateurs qui veulent modifier leur terminal Android, installer des ROM ou encore simplement installer des drivers USB sur l'ordinateur.

**Boot**

Le boot désigné la première étape de démarrage d'un système informatique. Sur les appareils mobiles Android, on parle de boot pour qualifier le démarrage du système, du recovery custom, du mode download ou encore du mode bootloader. Par extension, le terme Reboot veut dire redémarrage.

**Android Studio**

Android Studio est l’environnement de développement intégré (IDE) officiel pour les appareils Android. Il prend en charge Windows, Mac et Linux, est basé sur le logiciel IntelliJ IDEA de JetBrains et utilise Java, Cotlin et C ++. Le logiciel gratuit peut être téléchargé à partir d’Android Studio [site officiel](https://developer.android.com/studio) et contient des fonctionnalités qui incluent un éditeur de mise en page riche et une intégration intégrée avec Google Cloud. Outre Android Studio, les applications Android peuvent être conçues sur Visual Studio, AIDE, Eclipse, Droid4X, Bluestacks, etc.

**Versions Android**

Les systèmes d’exploitation Android ont vu de nombreuses versions différentes depuis son premier lancement, qui retracent une évolution substantielle au fil des ans. Les versions Android suivantes ont été publiées par Google.

# V Bibliographie

## 5.1 Github

<https://zone.ni.com/reference/fr-XX/help/371361R-0114/lvconcepts/glossary/>

## 5.2 Labview

<https://www.youtube.com/watch?v=rVwJ1mIwuqM>

<https://www.myandroidsolutions.com/2012/07/20/android-tcp-connection-tutorial/>

<https://www.ni.com/fr-fr/support/documentation/supplemental/06/basic-tcp-ip-communication-in-labview.html>

<https://www.youtube.com/watch?v=iiLmwBGUsyU>

<https://www.youtube.com/watch?v=LMtdLqmRnVU>

<https://www.youtube.com/watch?v=L3zirD3axCk>

<https://www.youtube.com/watch?v=29y4X65ZUwE>

<https://zone.ni.com/reference/fr-XX/help/371361R-0114/lvconcepts/glossary/>

## 5.3 Android studio

<https://www.youtube.com/watch?v=IibybM4oM1w&t=72s>

<https://www.youtube.com/watch?v=6pT112_KGkc>

<https://www.youtube.com/watch?v=IibybM4oM1w>

<https://www.youtube.com/watch?v=9Rwopuah2Q0>

# VI Résumé