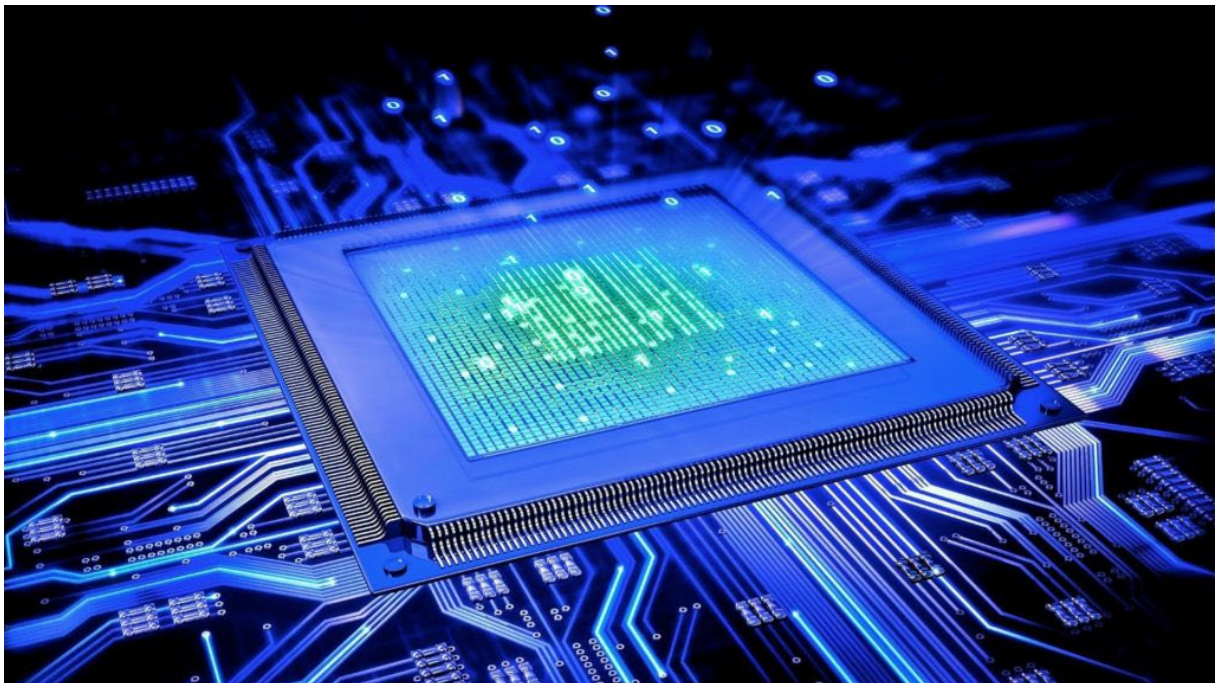


Mise en route des outils en phase industrielle électronique



Mise en route du projet

V2.0.0

Table des matières

Objectif de ce document	3
Création d'un compte GitHub	3
Installation et connexion à GitHub Desktop	3
Création du repository de votre projet	4
Clonage de votre repository GitHub sur votre ordinateur	6
Description de votre nouveau repository	7
Nomenclature des fichiers	8
Champ lexical de Git	8
Préparation de l'environnement Altium	9
Création ou réactivation de votre compte Altium	9
Connection au workspace de l'EMF	11
Créer votre nouveau projet Altium dans votre repository local	11
Paramètres du projet	12
Mettre à jour votre repository en ligne	13
Sémantique de version	14
Son incrémentation	14
Les spécifications de cette sémantique	15
Le numéro de version d'une release	15
Diagramme du versionning	15
Historique de version de ce document	Erreur ! Signet non défini.

Objectif de ce document

Cette démarche a pour but d'introduire les outils utilisés lors de la PHIE et de mettre en place le projet que vous devrez réaliser.

Un court champ lexical du langage Git et disponible en fin de document si nécessaire.

Création d'un compte GitHub

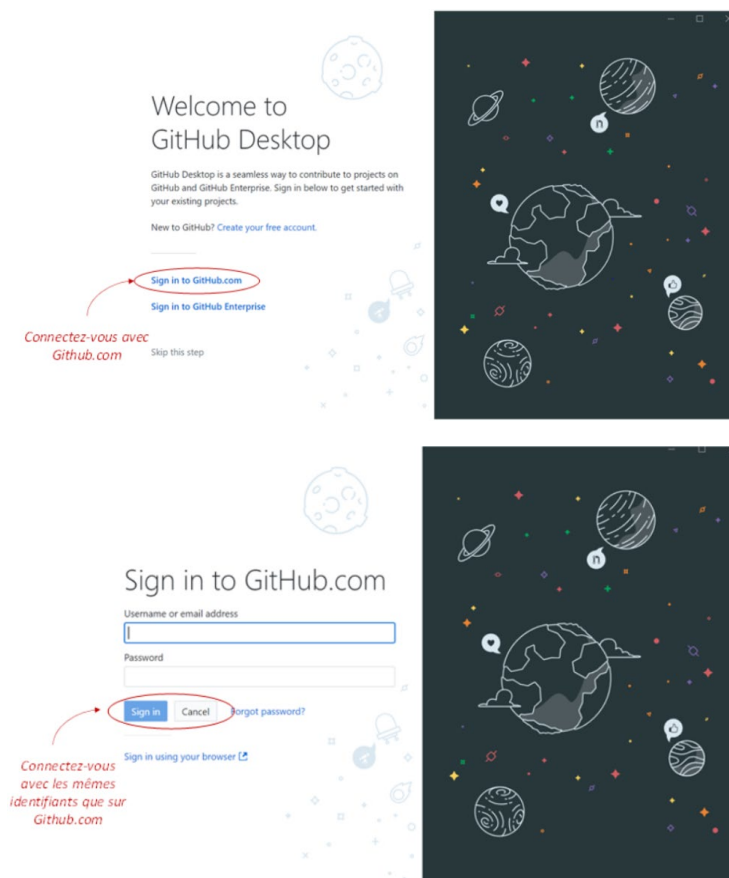
Accédez au site Github, www.github.com, puis créez-vous un compte en respectant les points suivants :

- Utiliser votre adresse **e-mail de l'EMF**.
- Utilisez votre **nom d'utilisateur de l'EMF** tout en **minuscule**. Si ce nom d'utilisateur est déjà utilisé ajoutez 01, si déjà utilisé incrémentez à 02, si déjà utilisé incrémentez à 03 et ainsi de suite jusqu'à trouver le nom d'utilisateur libre.
Par exemple pour monsieur Babakar Olof cela donne olofb, si déjà utilisé olofb01, si également déjà utilisé olofb02 et ainsi de suite.

Une fois votre compte créé, envoyez votre nom d'utilisateur GitHub dans le **canal Général de l'équipe Teams EMF-GT-PHIE**. Votre enseignant se chargera de vous inclure dans l'organisation **emf-elec-2x-2y** correspondant à l'année en cours.

Installation et connexion à GitHub Desktop

Pour ce faire, télécharger Github Desktop (si pas déjà disponible sur votre ordinateur), <https://desktop.github.com>, puis suivez les instructions ci-dessous :

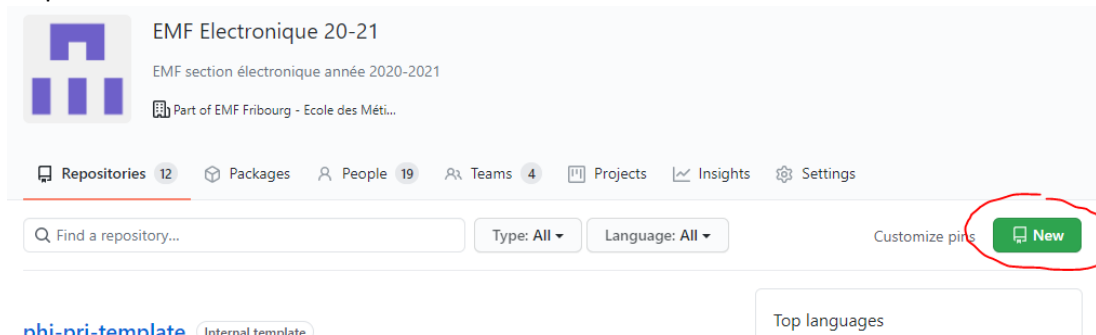


Création du repository de votre projet

Cette étape va créer un repository sur le serveur GitHub pour lequel vous serez administrateur. Lors de la création vous allez utiliser un template qui intégrera toute la structure du projet.

Pour créer un nouveau repository depuis l'interface web il faut :

1. Se rendre à l'adresse de l'organisation où vous voulez le créer, ici <https://github.com/emf-elec-2x-2y>.
2. Cliquer sur le bouton vert **New**



3. Entrer les informations nécessaires à la création du repository :
 - a. Sélectionner le template à utiliser, ici **emf-elec-2x-2y/altium-prj-template**
 - b. Sélectionner le propriétaire (owner) du repository, ici **emf-elec-2x-2y**
 - c. Introduire le nom de votre nouveau repository en respectant la nomenclature **NomDuProjet-NomDuCréateur** tout en **minuscule**.
 - i. **NomDuProjet** : Nom du projet/capteur/composant du développement en question.
 - ii. **NomDuCréateur** : Les trois premières lettres de votre nom de famille.
Par exemple pour le projet **Tic Tac Toe** réalisé par Monsieur Babakar **Olof**, le nom du repository sera **tictactoe-olo**.
 - d. Entrer une brève description du projet, par exemple « **Développement d'une carte électronique du jeu Tic Tac Toe pour la promotion du métier d'électronicien par Babakar Olof** ».
 - e. Sélectionner le type de visibilité, ici **Internal**. Pour information :
 - i. **Public** : Tout le monde peut le voir depuis internet en lecture seule et on peut choisir qui a des droits plus élevés. **NE PAS UTILISER POUR L'ECOLE SANS AUTORISATION PREALABLE**.
 - ii. **Internal** : Seulement les gens qui font partie de l'organisation du repository, ici emf-elec-2x-2y, peuvent avoir accès en lecture seule. On peut également décider à qui donner des droits plus élevés. **RECOMMANDE DE MANIÈRE GENERALE**
 - iii. **Private** : Personne ne le voit à l'exception de son créateur et des super-administrateur (vos enseignants). On peut également décider à qui donner des droits sur le repository. **A USAGE PERSONNEL MAIS VOUS LE PERDREZ EN QUITTANT L'ECOLE**.

4. Il ne vous reste plus qu'à cliquer sur le bouton vert Create repository :

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.


 emf-elec-21-22/tic-tac-toe-template ▾

☐ Include all branches

Copy all branches from emf-elec-21-22/tic-tac-toe-template and not just the default branch.

Owner *

Repository name *

 emf-elec-21-22 ▾ / prm-ttt-olo ✓

Great repository names are short and memorable. Need inspiration? How about [ideal-sniffle?](#)

Description (optional)

Développement d'une carte électronique du jeu Tic Tac Toe pour la promotion du métier d'électronicien par Bz

☐  Public

Anyone on the internet can see this repository. You choose who can commit.

☒  Internal

EMF Fribourg - Ecole des Métiers de Fribourg [enterprise members](#) can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

Create repository

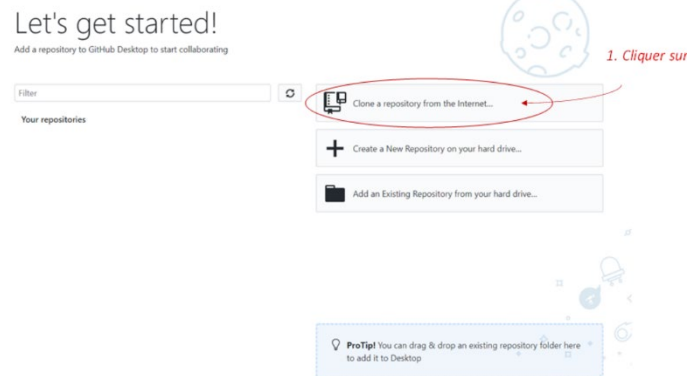
A partir de cet instant le repository existe sur le serveur de Github et vous pouvez y accéder depuis l'interface web mais également depuis l'outil GitHub Desktop qui permet de le cloner sur votre ordinateur.

Clonage de votre repository GitHub sur votre ordinateur

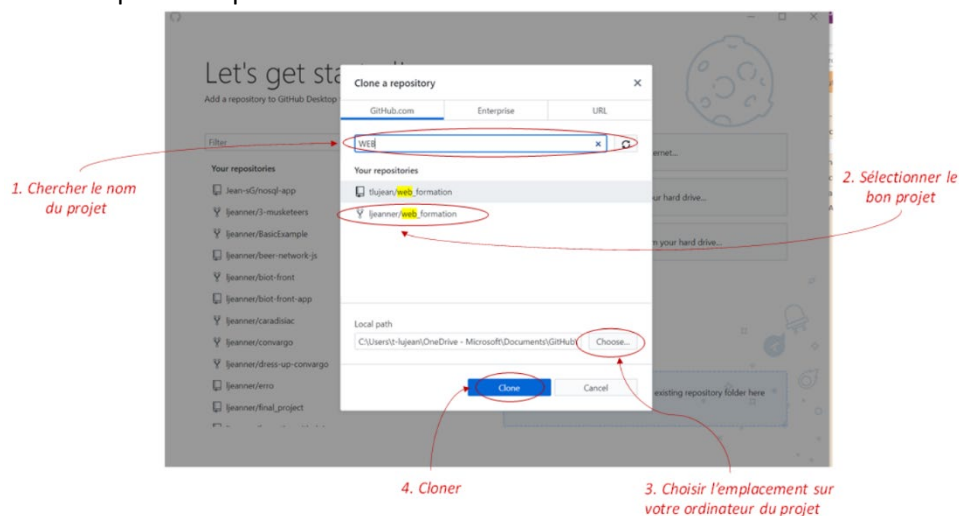
Cette étape permet d'avoir un clone de votre nouveau repository sur votre ordinateur. On appelle bien un clone et non pas une simple copie car un lien existe entre l'état de votre repository sur votre machine et son état sur le serveur de GitHub.

Après avoir ouvert GitHub desktop il vous faut :

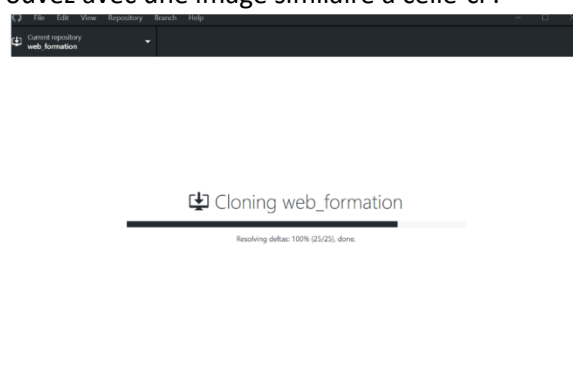
1. Cliquer sur le bouton **Clone a repository from the internet.**



2. Sélectionner le repository **projet-nom** que vous venez de créer dans la liste **Your repositories**. Si celui-ci n'est pas affiché vous pouvez le rechercher dans la barre de recherche. Choisissez ensuite l'emplacement sur votre ordinateur où sauvegarder le projet. Il est recommandé de créer un dossier **c:/workspace/NomUtilisateur** dans lequel travailler afin de s'affranchir des potentiel problème de réseau.



3. Une fois que tout est prêt cliquez sur le bouton bleu **Clone**. Si tout se passe correctement vous devriez vous retrouver avec une image similaire à celle-ci :



Description de votre nouveau repository

En observant votre repository maintenant cloné sur votre machine, vous pouvez voir différent dossier permettant l'organisation de votre projet de la manière suivante :

- **cdc** (Cahier Des Charges)
 - Contient le cahier des charges du projet ainsi que toutes les informations nécessaires à sa compréhension.
 - A l'intérieur de ce dossier les sous-dossiers et fichiers sont géré de manière personnalisée.
- **fab** (Fabrication)
 - Contient les fichiers de fabrication du projet. Ce répertoire ne contient que des fichiers .pdf ou .zip. Tous les fichiers sont renommés en respectant la sémantique de de version.
- **fwr** (Firmware)
 - Contient toutes les informations nécessaires au développement du firmware du système à micro.
 - Ce dossier contient un fichier gitignore spécialement conçu pour contenir un projet MPLABX.
- **mec** (Mécanique)
 - Contient toutes les infos échangées au sujet du projet (mails et autres notes)
 - A l'intérieur de ce dossier les sous-dossiers et fichiers sont géré de manière personnalisée.
- **pcb** (Printed Circuit Board)
 - Contient toutes les informations nécessaires à la construction du PCB (schéma, layout, BOM...)
- **rpt** (Rapport Technique)
 - Contient le rapport technique ainsi que tous les renseignements que l'on a jugé utile de collecter pour le projet.
 - Correspondances
 - Article sur un sujet du projet
 - Exemples, images, photos
 - Datasheet et/ou raccourcis vers Datasheets
 - etc...
 - A l'intérieur de ce dossier, des sous-dossiers et fichiers sont déjà présent et doivent être utilisé pour la sauvegarde de différents documents
- **sim** (Simulation)
 - Contient toutes les informations à de la simulation ou émulation du système à micro.
- **swr** (Software)
 - Contient les fichiers en langage objet destiné au PC gérant le système à microcontrôleur.
 - Ce dossier contient un fichier gitignore spécialement conçu pour contenir un projet Visual Studio.

Nomenclature des fichiers

Tous les fichiers d'un projet doivent être nommés en minuscule, les séparations entre les mots sont des tirets et doivent respecter la nomenclature suivante :

nomduprojet-xxx-prefixe-groupe-type
--

- **nomduprojet** : Le nom du projet
- **xxx** : Correspond aux 3 premières lettres de votre nom de famille
- **groupe** : Correspond au groupe de fichier auquel le fichier appartient
 - **pcb** : Pour les fichiers en lien avec le PCB
 - **rpt** : Pour les fichiers en lien aux rapports
 - **fwr** : pour les fichier en lien au firmware
 - **cdc** : Pour les fichiers en lien au cahier des charges
 - **fab** : Pour les fichiers en lien à la fabrication
 - **mec** : Pour les fichiers en lien à la mécanique
 - **sim** : Pour les fichiers en lien à la simulation
 - **swr** : Pour les fichiers en lien au software
- **type** : Correspond au type de fichier dont voici quelques exemples
 - cahier-des-charge
 - developpement
 - presentation
 - journal
 - bom
 - gerber
 - lib-i2c
 - ...

Champ lexical de Git

Branch : permet de travailler sur différentes versions d'un projet en même temps.

Clone : permet de télécharger sur son ordinateur le projet.

Commit : sur GitHub, les modifications enregistrées apportées à un projet sont appelées commit. Chaque commit regroupe plusieurs modifications avec le même message de validation. Ce message est une description destinée à « expliquer » les modifications.

Fork : permet à un utilisateur de créer un nouveau projet copié lui appartenant, ses modifications ne seront effectives que sur son projet.

Pull : Cette action permet de mettre à jour les informations en provenance des données stockées sur le serveur GitHub. Cela permet simplement de mettre à jour votre dossier sur votre ordinateur s'il est plus ancien que celui sur GitHub.

Push : Cette action transfère le commit depuis votre machine vers le serveur GitHub où les informations seront stockées. Cela permet simplement de mettre à jour le dossier sur GitHub s'il est plus ancien que celui sur votre machine.

Repository : On peut imaginer un repository en tant que dossier de projet stocké sur le serveur de GitHub. Il contient tous les sous-dossiers et fichiers que vous aurez sauvegardé pour le projet.

Préparation de l'environnement Altium

Quatre étapes sont nécessaires à la gestion de votre environnement de développement de PCB :

1. Créer ou réactiver votre compte Altium.
2. Vous connecter au workspace de l'EMF.
3. Créer un nouveau projet dans votre repository.
4. Adapter les paramètres de votre nouveau projet.
5. Mettre à jour vos modifications dans votre repository en ligne.

Création ou réactivation de votre compte Altium

Pour créer votre compte Altium, allez sur le lien suivant :

<https://www.altium.com/solutions/academic-programs/student-licenses>

Vous devez entrer :

- Votre **nom** et **prénom**.
- Courriel qui doit absolument contenir le domaine « **studentfr.ch** ». Vous ne pouvez pas mettre votre adresse e-mail personnelle.
- Numéro de **téléphone**
- Nom de l'université : **Ecole des Métiers de Fribourg**
- Pays : **Switzerland**
- Affiliation : **Student**

Après avoir cliqué sur « Submit », vous allez recevoir un courriel où vous devez activer la licence en donnant un mot de passe.

Ce processus peut prendre jusqu'à **24 heures**.

Free Enrollment for Altium Education

Please, complete this form to access curriculum, license and certification.

First Name*

Prénom


Last Name*

Nom

Accredited Email Address (EDU/University)*

Prenom.nom@studentfr.ch

Phone Number*

 +41 78 123 45 67

University Name*

Ecole des Métiers de Fribourg

Country/Region*

SWITZERLAND

SWITZERLAND

EDU/University Affiliation*

Student

Student

☐ Acknowledging Altium's [Privacy Policy](#), I consent that Altium processes my Personal Data to send me communications, including for marketing purposes, via email and to contact me by phone.

Submit

Vous devez confirmer votre e-mail et vous recevez l'email suivant où il faut cliquer sur « Activate AltiumLive »:

About Your Altium Designer Student License

Hi Rémi,

Thank you for choosing Altium Designer. This email contains your license details and information about getting started with your new Altium software.

If you are already familiar with the Altium installation and activation process, simply scroll down this email to find your license information.

Step 1: Activate AltiumLive

AltiumLive is where you access the Altium online ecosystem. If you have not already done so, you will need to activate your AltiumLive account before you can continue.

[Activate AltiumLive](#)

Step 2: Download Altium Designer

Installers for Altium products are available from the downloads page. You will need to login with your AltiumLive account you activated in the previous step to download the installers.

[Altium Downloads](#)

Step 3: Install and Activate Software

Double click the installer application, login with your previously created AltiumLive account, and follow the on screen instructions to install. Further details are available here:

[Installing Altium Designer](#)

Licensing for Altium software depends on your specific license type. You can find additional details regarding your license type here:

Introduisez ensuite un mot de passe et cliquez sur « NEXT »

Pour utiliser le logiciel, vous devez vous loguer avec votre compte.

La licence est valable 6 mois, vous devez renouveler la manipulation tous les 6 mois.

Set Up Your Altium Account

Create a Password

Password requirements:

- Must contain at least 8 characters
- Must contain at least one number and one uppercase character
- Must use only English characters
- Cannot contain spaces

Email: jonas.bruegger@studentfr.ch

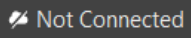
New Password:

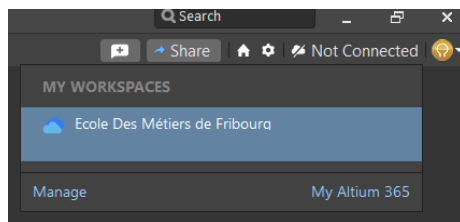
Confirm Password:

Please use credentials above during installation process.

[NEXT](#)

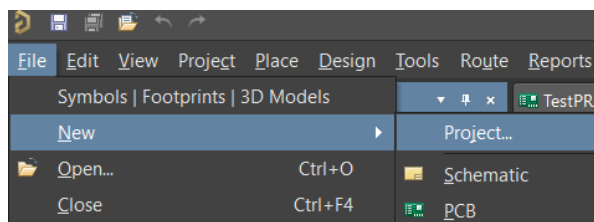
Connection au workspace de l'EMF

Cliquer sur  puis sélectionner « Ecole Des Métiers de Fribourg »

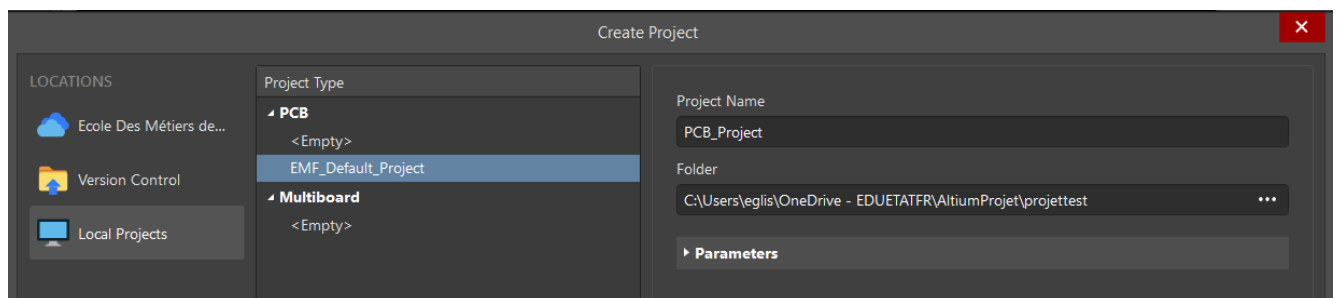


Créer votre nouveau projet Altium dans votre repository local

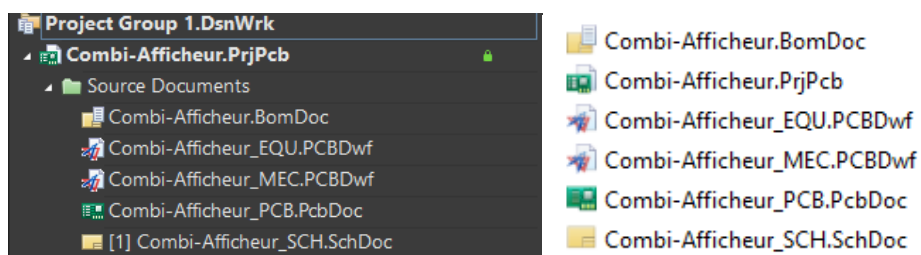
1. Cliquer sur « File → New → Project... »



2. Sélectionner « Local Projects » et « EMF_Default_Project » en mettant le nom du projet et son emplacement sur le disque (**qui est le dossier « PCB » de votre nouveau repository clôné**).

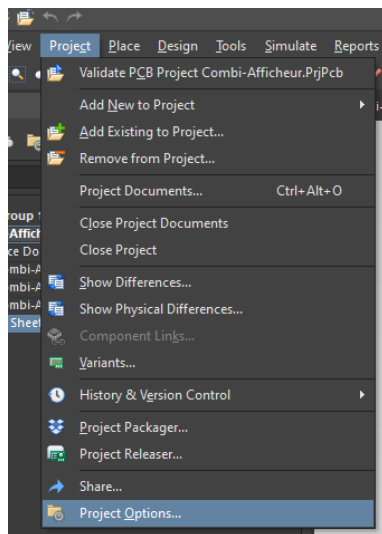


A ce moment-là, tous les fichiers du projet seront renommés automatiquement avec le nom du projet. Dans cet exemple, le projet s'appelle « Combi-Afficheur ».

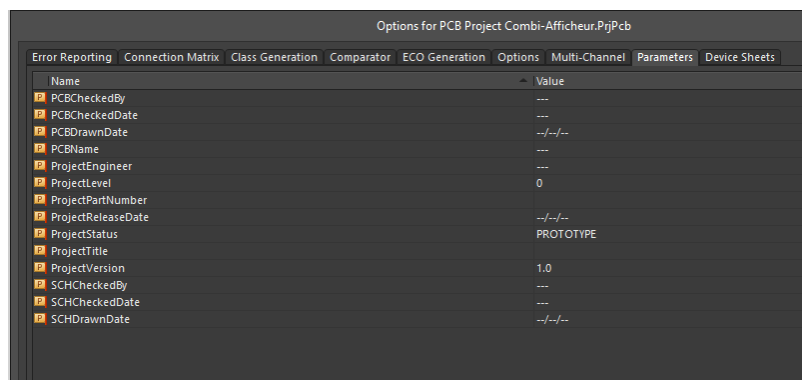


Paramètres du projet

1. Vous pouvez accéder à ces paramètres dans l'onglet « Project »



2. Ensuite sous « Parameters » vous accédez à tous les paramètres du projet qu'il faut remplir



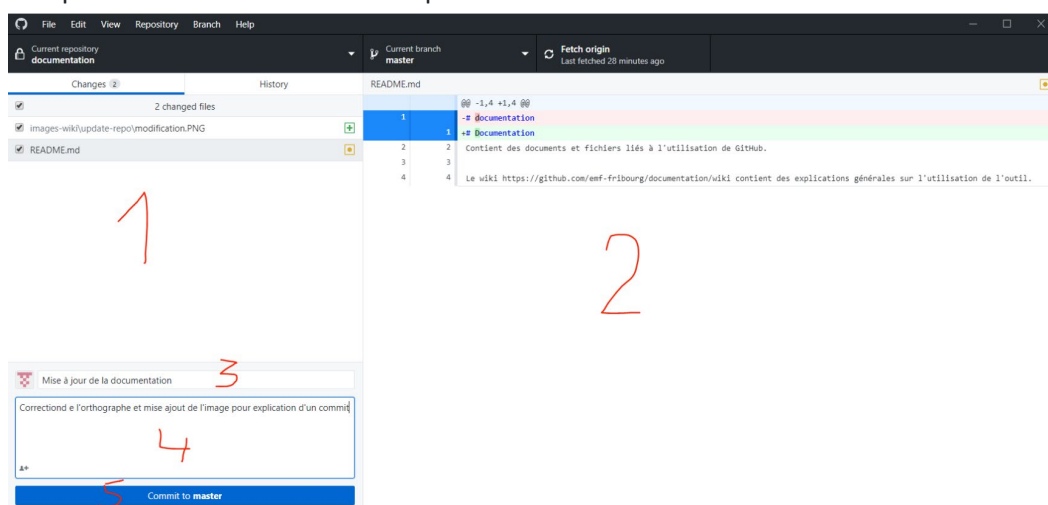
Définition des paramètres

PCBCheckedBy	Nom de la personne qui a contrôlé le PCB
PCBCheckedDate	Date de contrôle du PCB
PCBDrawnDate	Date de réalisation du PCB
PCBName	Nom du PCB
ProjectEngineer	Nom du concepteur
ProjectLevel	Version patch du projet
ProjectPartNumber	Numéro de fabrication du PCB
ProjectReleaseDate	Date de libération du projet
ProjectTitle	Nom du projet
ProjectVersion	Version majeure.mineure du projet
SCHCheckedBy	Nom de la personne qui a contrôlé les schémas
SCHCheckedDate	Date de contrôle des schémas
SCHDrawnDate	Date de réalisation des schémas

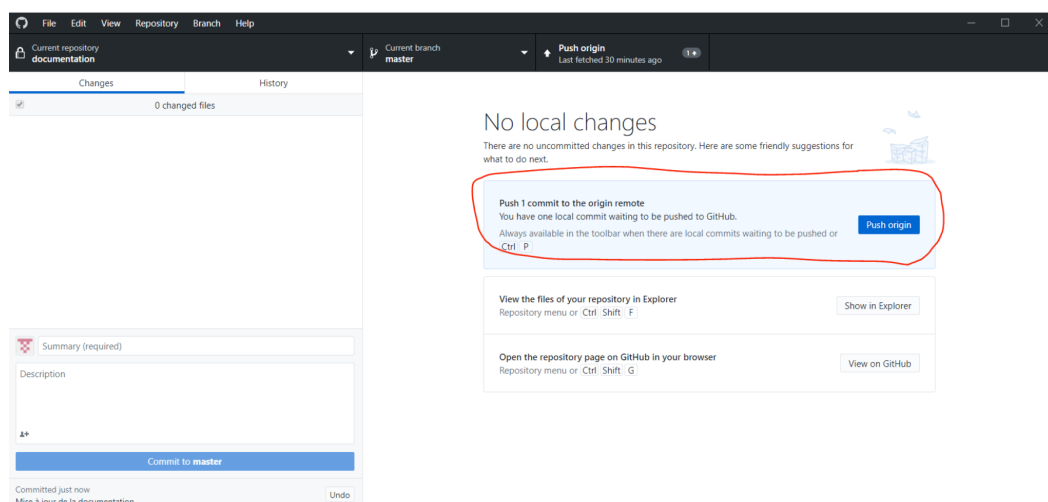
Mettre à jour votre repository en ligne

Lorsque des fichiers sont modifiés sur la PC et que vous désirez les mettre à jour également en ligne, après avoir ouvert GitHub Desktop, la démarche est la suivante :

1. L'interface montre
 - En 1: Les fichiers différents entre votre dossier local et le repository en ligne.
 - En 2: Les différences dans le fichier sélectionné au point 1.
2. Il faut contrôler que les différences sont bien celles attendues.
3. Si tout est correct il est nécessaire de donner un titre du commit qui sera créé au point 3.
4. Ajouter un commentaire détaillé au point 4 qui décrit les modifications apportées. Ceci permet par la suite de facilement revenir sur un commit qui aurait pu, par exemple, intégrer un nouveau bug.
5. Cliquer sur **Commit to Master** au point 5.



Une fois le commit créé, il suffit d'effectuer un push pour que celui-ci soit transféré sur le repository en ligne. GitHub vous empêchera d'effectuer un push si il y a un problème de synchronisation entre le repository et le commit que vous essayez de pusher. Dans ce cas-là il vous sera demandé de gérer le conflit afin de faire le choix de l'état de la version finale des fichiers problématique.



Chaque commit est référencé et il est possible à n'importe quel moment de revenir à un état défini par un commit si nécessaire.

Sémantique de version

Tout au long du développement de votre projet vous allez devoir versionner les différents fichiers que vous allez créer. La version des différents éléments du projet, cahier des charges, pcb, rapport technique, firmware, software et mécanique doivent suivre la numérotation suivante :

MAJEUR.MINEUR.CORRECTIF

- **Majeur** étant égal à la version du cahier des charges.
- **Mineur** étant égal à la version du PCB
- **Correctif** étant variable pour chaque modification en interne des différents éléments du projet précités.

Son incrémentation

Il faut incrémenter :

- Le numéro de version **MAJEUR** lorsqu'il y a des changements du cahier des charges qui implique une modification du PCB.
- Le numéro de version **MINEUR** lorsqu'il y a des modifications de schéma ou de layout qui nécessitent la création de nouveau fichiers GERBER du PCB.
- Le numéro de version de **CORRECTIF** lorsqu'il y a des corrections interne à un élément du projet.

De manière plus détaillée :

- La version du cahier des charges
 - Son numéro ****Majeur**** est incrémenté lorsqu'un changement implique une modification du PCB.
 - Possède toujours un numéro ****Mineur**** égal à zéro (Majeur.0.Correctif).
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.
- La version du PCB
 - Son numéro **Majeur** est toujours égal au numéro Majeur du cahier des charges auquel il se rapporte.
 - Son numéro **Mineur** est incrémenté lorsque des modifications impliquent une modification du PCB.
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.
- La version du Rapport technique
 - Son numéro **Majeur** est toujours égal au numéro Majeur du cahier des charges auquel il se rapporte.
 - Son numéro **Mineur** est toujours égal au numéro Mineur du PCB auquel il se rapporte.
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.

- La version du firmware
 - Son numéro **Majeur** est toujours égal au numéro Majeur du cahier des charges auquel il se rapporte.
 - Son numéro **Mineur** est toujours égal au numéro Mineur du PCB auquel il se rapporte.
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.
- La version du software
 - Son numéro **Majeur** est toujours égal au numéro Majeur du cahier des charges auquel il se rapporte.
 - Son numéro **Mineur** est toujours égal au numéro Mineur du PCB auquel il se rapporte.
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.
- La version de la mécanique
 - Son numéro **Majeur** est toujours égal au numéro Majeur du cahier des charges auquel il se rapporte.
 - Son numéro **Mineur** est toujours égal au numéro Mineur du PCB auquel il se rapporte.
 - Son numéro ****Correctif**** est incrémenté lorsque des corrections rétro compatibles sont appliquées.

Les spécifications de cette sémantique

- Un numéro de version standard doit prendre la forme X.Y.Z où X, Y et Z sont des entiers non négatifs et NE DOIVENT PAS être préfixés par des zéros.
- Une fois qu'une version est définie, le contenu ne doit pas être modifié. Toute modification doit être publiée dans une nouvelle version.
- L'identifiant de version majeure zéro (0.0.z) est destiné au développement initial. Tout ou une partie peut être modifié à tout moment. Cette version ne doit pas être considérée comme stable.
- La version 1.0.0 définit la première version stable du cahier des charges. La façon dont le numéro de version est incrémenté après cette publication est dépendante de ses évolutions.
- La version 1.1.0 définit la première version stable du PCB lié à la version 1.0.X du cahier des charges auquel il se rapporte. La façon dont le numéro de version est incrémenté après cette publication est dépendante de ses évolutions.

Le numéro de version d'une release

Une nouvelle release doit porter le numéro de version suivant :

- Numéro **Majeur** égal au numéro Majeur du cahier des charge auquel elle se rapporte.
- Numéro **Mineur** égal au numéro Mineur du PCB auquel elle se rapporte.
- Numéro **Correctif** égal au numéro Correctif du rapport technique auquel elle se rapporte. Si la numérotation des éléments du projet a été respecté, cela doit correspondre exactement au dernier numéro de version du rapport technique. Cela implique qu'une release ne peut pas être effectuée sans mettre à jour le rapport technique.

Diagramme du versionning

Afin de rendre les choses un peu plus clair un diagramme de versionning est présentée à la page suivante permettant de suivre les règles de la sémantique plus aisément.

Vous pouvez y accéder de manière informatique et en plus grand à l'adresse <https://github.com/emf-elec-21-22/documentation/blob/master/images-wiki/release/e-prj-versioning.png>

