



Documentation Latex

Bibliothèque Utils

Nicolas Le Guerroué

14 mai 2022
79 pages

TABLE DES MATIÈRES

	Page
1 Préambule	6
1.1 Conventions	6
2 Introduction	8
2.1 Présentation	8
3 Installation	10
3.1 Installation des outils	10
4 Architecture	12
4.1 Organisation du projet	12
5 Compilation	15
5.1 Compilation avec un terminal	15
5.2 Compilation avec VSCode	19
6 Mise à jour des bibliothèques	24
7 Création d'un nouveau projet	27
8 Fusion de projets	28
8.1 Conventions	28
9 Bibliothèque Badges	29
9.1 Création de badges avec des couleurs	29
10 Bibliothèque Bibliography	30
10.1 Format de la bibliographie	30
10.2 Affichage de la bibliographie	30

11 Bibliothèque Colors	32
11.1 Afficher du contenu en couleur	32
11.2 Liste des couleurs disponibles	32
12 Bibliothèque Electronic	34
12.1 Création de chronogrammes fixes	34
12.2 Création de chronogrammes flottants	35
12.3 Création de schémas électriques	36
13 Bibliothèque Figures	38
13.1 Création d'une figure centrée	38
13.2 Création d'une figure non centrée	38
13.3 Affichage de la liste des figures	38
14 Bibliothèque Fonts	40
14.1 Ajout d'icônes	40
15 Bibliothèque Glossaries	41
15.1 Ajout d'une définition au glossaire	41
15.2 Affichage du glossaire	41
16 Bibliothèque Graphic	43
16.1 Affichage d'un graphique 2D avec insertion des données depuis un fichier txt (csv) .	43
16.2 Affichage d'un graphique 2D avec insertion des données depuis une liste de points .	44
16.3 Affichage d'un graphique 2D avec insertion des données depuis une équation	45
16.4 Affichage de deux graphiques	46
17 Bibliothèque Header	48
17.1 Mise en forme de la page de garde avec une image	48
17.2 Mise en forme de la page de garde sans image	48
17.3 Mise en forme de la page des parties	48
17.4 Ajout d'un trait entre l'en-tête et le corps de la page	49
17.5 Ajout d'un trait entre le corps de la page et le bas de page	49
17.6 Définition de la présentation globale des pages	49
17.7 Redéfinition des titres des chapitres	49
17.8 Mettre le document en pleine page	50
17.9 Récupérer le chapitre courant	50
18 Bibliothèque Images	51
18.1 Définition de l'espace de nommage	51
18.2 Ajout d'une image non-flottante	51
18.3 Ajout d'une image non-flottante avec une rotation	52
18.4 Ajout d'une image flottante	52
19 Bibliothèque Items	53
19.1 Création d'une liste	53

20 Bibliothèque Labels	54
20.1 Création de labels générique	54
20.2 Création de labels spécifiques	54
21 Bibliothèque Layout	56
21.1 Mise en gras	56
21.2 Mise en italique	56
21.3 Mise en gras et italique	56
21.4 Ajout d'un espace vertical	57
22 Bibliothèque Links	58
22.1 Paramétrage des liens et des méta-données	58
23 Bibliothèque Maths	59
23.1 Création d'une matrice 3*3	59
23.2 Création d'un vecteur à trois dimensions	59
23.3 Création d'un torseur à trois dimensions	59
24 Bibliothèque MessageBox	61
24.1 Création de boites de dialogues	61
25 Bibliothèque Nomenclature	62
25.1 Ajout d'un élément dans la nomenclature	62
25.2 Affichage de la nomenclature	63
26 Bibliothèque Object3D	65
26.1 Affichage d'un graphique 3D avec insertion des données depuis une équation	65
26.2 Affichage de sphères en 3D	65
27 Bibliothèque Pdf	66
27.1 Insertion d'un document PDF	66
27.2 Insertion d'un ensemble de pages d'un document PDF	66
28 Bibliothèque Programming	67
28.1 Affichage d'un code C/C++ avec titre	67
28.2 Affichage d'un code C/C++ sans titre	68
28.3 Affichage d'un code Python avec titre	68
28.4 Affichage d'un code Python sans titre	69
28.5 Affichage d'un code Bash avec titre	69
28.6 Affichage d'un code bash sans titre	69
29 Bibliothèque Quotes	70
29.1 Création d'une citation	70
30 Bibliothèque Tables	71
30.1 Création d'un tableau	71
30.2 Affichage de la liste des tables	72

31 Bibliothèque Theorems	73
31.1 Création d'une question	73
31.2 Création d'une reponse	73
31.3 Création d'une propriete	74
31.4 Création d'une proposition	74
31.5 Création d'une remarque	74
31.6 Création d'un exemple	74
31.7 Création d'une définition	75
31.8 Création d'une solution	75
32 Bibliothèque Titles	76
32.1 Titre de section	76
33 Bibliothèque Tree	77
33.1 Création d'une arborescence simple	77
33.2 Création d'une arborescence plus évoluée	77
33.3 Création d'une arborescence dans une figure	78
Liste des figures	79

SECTION 1

PRÉAMBULE

- ▶ Document réalisé en \LaTeX par Nicolas Le Guerroué
- ▶ Permission vous est donnée de copier, distribuer et/ou modifier ce document sous quelque forme et de quelque manière que ce soit.
- ▶ Version du 14 mai 2022
- ▶ Taille de police : 11pt
- ✉ nicolasleguerroue@gmail.com
- ▶ Dans la mesure du possible, évitez d'imprimer ce document si ce n'est pas nécessaire. Il est optimisé pour une visualisation sur un ordinateur et contient beaucoup d'images.

Versions

janvier 2021 Version de base avec la commande *make*
mai 2022 Version avec l'intégration à VSCode




Conventions

Les commandes à saisir sont dans des encadrés similaires :







```
sudo apt-get update
```

Exemple de commande

Parfois, ces encadrés contiendront des instructions qu'il faudra placer dans certains fichiers.

- Les fichiers sont indiqués par le repère  fichier
- Les dossiers sont indiqués par le repère  dossier
- Les logiciels sont indiqués par le repère  logiciel¹

1. Sont également concernés les paquets Linux et les bibliothèques des langages

- Les adresses IP sont indiquées par le repère  Adresse IP
- Les adresses MAC sont indiquées par le repère  Adresse MAC
- Les liens sont indiqués par le repère  Lien
- Les broches génériques des composants sont indiquées par le repère  Broche et se scindent en deux parties :
 - Les broches d'entrée par  Broche
 - Les broches de sortie par  Broche
-

SECTION 2

INTRODUCTION

Présentation

Ce document a pour but de présenter les fonctionnalités de la bibliothèque Utils, qui n'est qu'un regroupement de bibliothèques pour simplifier l'utilisation de Latex.

Voici les bibliothèques disponibles :

- Badges
- Bibliography
- Colors
- Debug
- Electronic
- Figures
- Fonts
- Glossaries
- Graphics
- Header
- Images
- Index
- Items
- Labels
- Layout
- Links
- Lipsum

- Maths
- MessageBox
- Nomenclature
- Objects3D
- Parts
- Pdf
- Programming
- Quotes
- Tables
- Theorems
- Titles
- Tree

SECTION 3

INSTALLATION

Installation des outils

Installation détaillée

Latex est un logiciel assez volumineux¹ mais l'installation complète ne nécessite pas d'ajout de paquet supplémentaire. Il est disponible dans les dépôts **Debian/Ubuntu** avec les commandes suivantes² :

```
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get install texlive-full
```

Installation de Latex

La commande suivante permet de gérer Latex en français.

```
sudo apt-get install texlive-lang-european
```

Installation des langues

Remarque

L'ensemble des outils présentés est optimisé pour une utilisation avec le logiciel Visual Studio Code

Installons donc le logiciel VSCode :

```
sudo snap install code
```

Installation de Visual Studio Code

Le logiciel okular permettra de visualiser les fichiers PDF.

```
sudo apt-get install -y okular
```


Installation du logiciel Okular

1. Environ 1.5Go dans les dépôts Debian/Ubuntu
2. Il faut saisir la commande dans un terminal

L'installation de PHP permettra de générer le code d'autocomplétion pour VSCode ainsi que certaines fonctionnalités pour le traitement des fichiers à compiler.

```
sudo apt-get install -y php
```

Installation du module PHP

Enfin, installons  **Aspell** pour la correction orthographique.

```
sudo apt-get install -y aspell
```

Installation de aspell

Installation rapide

Le fichier  **make** peut installer tous les outils nécessaires.

Il faut auparavant donner les droits d'exécution au fichier en saisissant dans un terminal :

```
chmod +x make
```

Don des droits

Puis dans le même terminal :

```
./make --install
```

Don des droits

SECTION 4

ARCHITECTURE

Organisation du projet

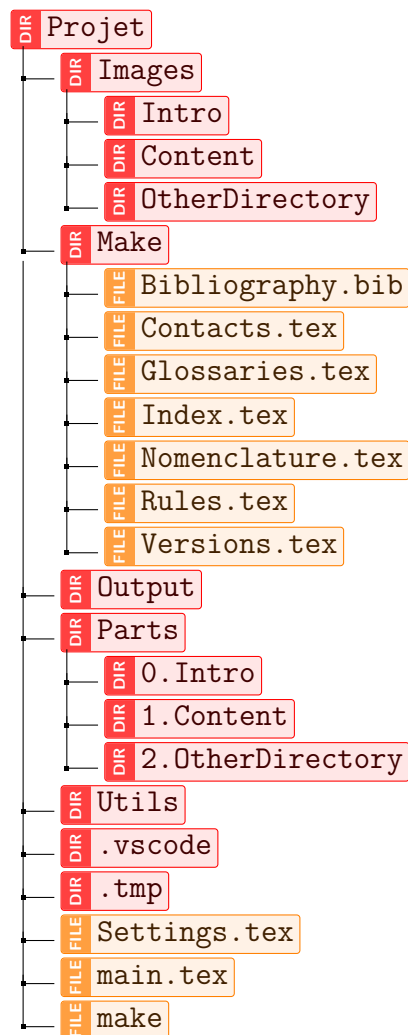


FIGURE 4.1 – Arborescence du projet

Chaque projet est constitué de 7 dossiers et de 3 fichiers situés à la racine du projet.

- Le dossier `DIR Images` contient l'ensemble des images du projet. Chaque image doit faire partie

de la même partie que son document source associé.

La gestion des emplacements des images est indiqué dans la section [LINK Définition de l'espace de nommage](#)

- ▶ Le dossier **DIR Make** contient les fichiers annexes du projet :
 - ▶ Le fichier **FILE Bibliography.bib** recense les bibliographies du projet.
Pour plus d'informations, consulter la section [LINK Bibliothèque Bibliography](#).
 - ▶ Le fichier **FILE Contacts.tex** est une page pour contacter l'auteur et contient les informations sur les droits et les licences du projet.
 - ▶ Le fichier **FILE Versions.tex** contient les différentes versions du projet.
Pour plus d'information, consulter la partie [LINK Ajout de version](#)
 - ▶ Le fichier **FILE Glossaries.tex** contient le glossaire.
Pour ajouter une définition, veuillez vous référer à la section [LINK Bibliothèque Glossaries](#)
 - ▶ Le fichier **FILE Index.tex** contient l'index.
 - ▶ Le fichier **FILE Nomenclature.tex** contient la structure de la nomenclature¹.
Pour ajouter un élément, consulter la section [LINK Bibliothèque Nomenclature](#)
 - ▶ Le fichier **FILE Rules.tex** contient les conventions pour le projet. Il peut contenir les types de commandes, les conventions de nommage du projet...
- ▶ Le dossier **DIR Output** contient les fichiers de compilation générés de manière automatique.
Vous n'aurez pas à modifier des fichiers à cette emplacement.
- ▶ Le dossier **DIR Parts** contient les différentes parties du projet. Il est possible de scinder son projet en grandes parties (Introduction, Chapitre1, Chapitre2, Conclusion), chaque dossier contenu dans le dossier **Parts** représente ces parties.

Dans chacun de ces dossiers, vous pouvez créer autant de fichier Latex que vous voulez, il seront compilés dans l'ordre alphabétique ou bien par ordre croissant si vous mettre un numéro au début du nom de fichier.

Pour nommer les dossiers dans le dossier **DIR Parts**, **il faut impérativement commencer le nom avec un numéro suivi d'un point**. Par exemple **DIR 0.Intro** puis **DIR 1.Content**.

Les espaces sont interdits dans les noms des dossiers mais les symboles `_` et `-` sont acceptés.

Pour chaque dossier crée dans le dossier **DIR Parts**, il faudra créer un dossier avec le même nom dans le dossier **DIR Images**, sous peine de voir une volée d'erreurs lors de la compilation.

1. Les unités et grandeurs physiques par exemple

- ▶ Le dossier **DIR** `Utils` contient les bibliothèques du projet. Le fichier **FILE** `Utils.sty` est généré dynamiquement, toute écriture manuelle sera écrasée à la prochaine compilation.
- ▶ Le dossier **DIR** `.vscode` (dossier caché) contient les fichiers des paramètres VSCode ainsi que quelques utilitaires :
 - ▶ Le fichier **FILE** `compileAllFiles.php` désactive la compilation pour l'ensemble des dossiers contenus dans le dossier **DIR** `Parts`.²
 - ▶ Le fichier **FILE** `compileAnyFile.php` autorise la compilation des dossiers contenus dans le dossier **DIR** `Parts`.³
 - ▶ Le fichier **FILE** `generateSnippets.php` créer un fichier de snippets VScode afin de gérer l'autocomplétion VSCode. **Ce fichier doit obligatoirement être appelé depuis le fichier **FILE** `make`.**
 - ▶ Le fichier **FILE** `settings.json` contient les tâches exécutables VScode via un bouton⁴. Ces tâches permettent de lancer la compilation et les utilitaires du répertoire.
 - ▶ Le fichier **FILE** `task_template.sh` est un fichier en développement pour générer automatiquement le fichiers **FILE** `tasks.json`.
 - ▶ Le fichier **FILE** `task.json` est le fichier qui recense les tâches exécutables sous VSCode.
- ▶ Le dossier **DIR** `.tmp` (dossier caché) contient les fichiers temporaires utilisés pour la mise à jour de la bibliothèque **LIB** `Utils` depuis Git.

Et voici les trois fichiers situés à la racine :

- ▶ Le fichier **FILE** `Settings.tex` regroupe les paramètres de mise en page du projet.⁵
- ▶ Le fichier **FILE** `main.tex` est le fichier principal du projet. C'est dans ce fichier qu'on définit notamment :
 - ▶ Le choix de la classe du document.
 - ▶ Le titre de la page
 - ▶ Le choix de la disposition du document
 - ▶ Le choix de la présence des fichiers dans le dossier **DIR** `Make`, c'est à dire les fichiers personnalisables tels que la bibliographie, l'index, le glossaire...
- ▶ Le fichier **FILE** `make` est le fichier de compilation. **Vous n'aurez pas besoin de modifier ce fichier pour une utilisation classique**

2. Concrètement, cela revient à ajouter un point devant le nom du dossier pour que le compilateur ignore le dossier.

3. On retire les points devant les noms.

4. Cf. **LINK** `Les raccourcis clavier`

5. Cf. **LINK** `Ajout de version`

SECTION 5

COMPILATION

Don des droits

La compilation du projet se fait grâce au fichier `FILE make` situé à la racine du projet. Avant de faire la toute première compilation, il convient de rendre exécutable le fichier `FILE make` en saisissant la commande suivante :

```
chmod +x make
```

Don des droits d'exécution sur le fichier `FILE make`

L'ensemble des outils de compilation sont disponibles de deux façons :

- Via un terminal
- Via le logiciel VSCode

L'avantage du logiciel VSCode est qu'il fournit des raccourcis clavier et une interface graphique plus évoluée (boutons)

Compilation avec un terminal

Compilation classique

Une compilation classique a pour objectif de générer le fichier PDF de rendu, appelé `FILE main.pdf` et situé à la racine du projet.

La commande est la suivante :

```
./make --full
```

Compilation complète du projet

Lors de la compilation, plusieurs fichiers sont générés à la racine, dont :

- ▶ Le fichier `FILE .render_report.tex` (fichier caché) qui contient la première partie des fichiers journaux de compilation

- ▶ Le fichier `FILE.render_report_logs.tex` qui contient la seconde partie des fichiers journaux de compilation¹
- ▶ Une image *Part.png* qui affiche le nombre de ligne pour chaque fichier compilé contenu dans le dossier **Parts**

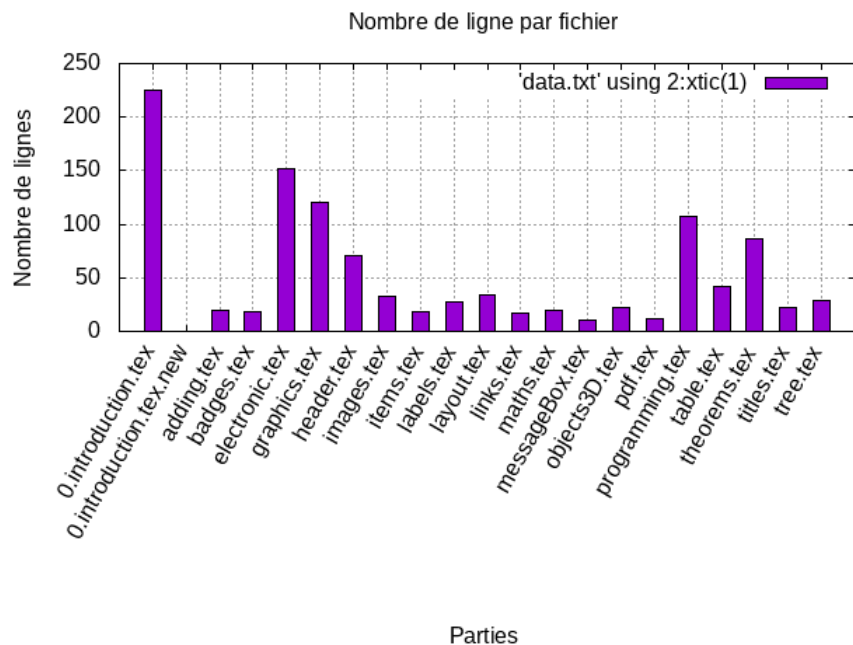


FIGURE 5.1 – Nombre de ligne pour les parties

- ▶ Une image *Utils.png* qui affiche le nombre de ligne pour chaque fichier contenu dans le dossier **Utils**

1. Les messages de compilation générés par la bibliothèque Utils sont situés dans ce fichier.

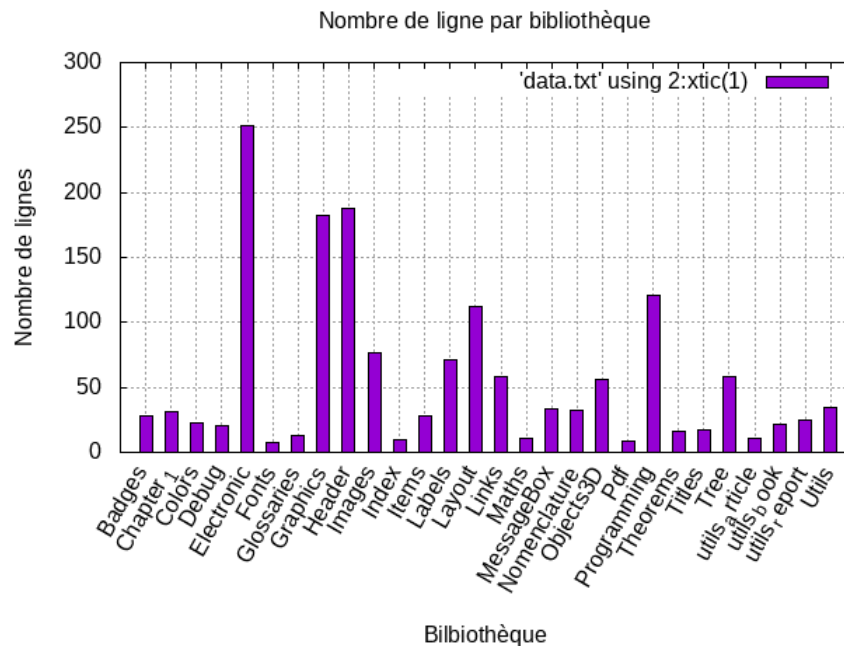


FIGURE 5.2 – Nombre de ligne pour les bibliothèques

- Un fichier `standlone.tex` qui contient l'ensemble du code latex. Ce fichier est en cours de développement.

Lors de la compilation, différents messages s'affichent :

```
>>> Messages :
>>> Utils : Babel package is loaded
>>> Utils - [MData] : title='Tutoriel Latex'
>>> Utils - [MData] : author(s)='Nicolas Le Guerroué'
>>> Utils - [MData] : subject='Bibliothèque Utils'
>>> Utils - [MData] : creator='Nicolas Le Guerroué'
>>> Utils - [MData] : keywords='Latex'
>>> Utils - [MData] : link colors='green'
>>> Utils - [MData] : bib links colors='blue'
>>> Utils - [MData] : link file colors='blue'
>>> Utils : Image 'Images/content/Part.png' [size=0.5,id 1] loaded !
>>> Utils : Image 'Images/content/Utils.png' [size=0.5,id 2] loaded !
>>> Utils : Image 'Images/content/check.png' [size=0.5,id 3] loaded !
>>> Utils : Image 'Images/content/check2.png' [size=0.5,id 4] loaded !
>>> Utils : Image 'Images/content/tux.png' [size=0.5,id 5] loaded !
>>> Utils : Image 'Images/content/tux.png' [size=0.5,id 6,angle=45] loaded !
>>> Utils : MessageBox 'Message' [id 1] created !
>>> Utils : MessageBox 'Message' [id 2] created !
>>> Utils : MessageBox 'Message' [id 3] created !

>>> Compilation terminée !
```

FIGURE 5.3 – Message d'ajout d'éléments de la bibliothèque Utils

```
>>> Warnings :
Package Utils Warning: Image 'Images/content/Parts.png' no loaded on input line
```

FIGURE 5.4 – Message d’avertissements

Vérification orthographique

En invoquant le paramètre `--check` avec la commande `make`, il est possible de faire une vérification orthographique avec le logiciel `aspell`. Enfin, si vous lancez la commande

```
./make --check
```

Vérification orthographique

Le fichier `make` vous demande si vous souhaitez corriger les fichiers contenus dans le dossier `Parts`.

```
(base) nico@nico-ThinkPad:~/Documents/Projets GIT/Latex$ ./make --check
Vérification orthographique du répertoire Parts...
>>> Dossier Parts/0.content en cours d'analyse !
Voulez-vous analyser le fichier Parts/0.content/0.introduction.tex ? (y/n)
```

FIGURE 5.5 – Vérification orthographique

Veuillez saisir `y` si vous souhaitez corriger le fichier indiqué. Ensuite, il ne vous reste plus qu'à être guidé par le logiciel `aspell`.

```
nico@nico-ThinkPad: ~/Documents/Projets GIT/Latex
Voici les bibliothèques disponibles :
\begin{items}{\blue}{\Circle}
\item Badges
\item Colors
\item Debug
\item Electronic
\item Fonts
\item Glossaries
\item Graphics
\item Header

1) blues                6) bleue
2) bluet                7) blé
3) bleu                 8) bue
4) blute                9) lue
5) bluté               0) boue
i) Ignorer              I) Ignorer tout
r) Remplacer            R) Remplacer tout
a) Ajouter              l) Ajouter minus.
b) Abandon              x) Sortir
```

FIGURE 5.6 – Commande de vérification orthographique

Les commandes sont à saisir au clavier ( pour ignorer le mot par exemple).

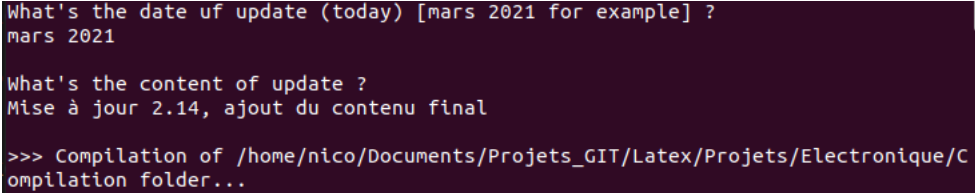
Ajout de version

Il est possible d'ajouter une version de projet en invoquant la commande suivante :

```
./make --version
```

Mise à jour Git

La date de la mise à jour vous sera demandée ainsi que le contenu de la mise à jour.



```
What's the date of update (today) [mars 2021 for example] ?
mars 2021


What's the content of update ?
Mise à jour 2.14, ajout du contenu final

>>> Compilation of /home/nico/Documents/Projets_GIT/Latex/Projets/Electronique/C
compilation folder...
```

FIGURE 5.7 – Ajout d'une version

Mise à jour de l'autocomplétion

Il est possible de mettre à jour l'autocomplétion sous VScode pour les bibliothèques Utils. En invoquant le paramètre **--snippet**, il est possible de générer le fichier VScode qui va ajouter l'autocomplétion.

Ce fichier est appelé  **output.snippet-code** et se situe dans le dossier  **.vscode**

```
./make --snippet
```

Mise à jour de l'autocomplétion

La commande se termine en affichant le nombre de commandes documentées :

Compilation avec VSCode

l'ensemble des commandes présentées précédemment sont disponibles sous VSCode.

Ouverture d'un projet

Il faut ouvrir un terminal à la racine du projet et lancer la commande :

```
code .
```

Ouverture de l'arborescence avec VScode

Le logiciel se lance avec une interface similaire :

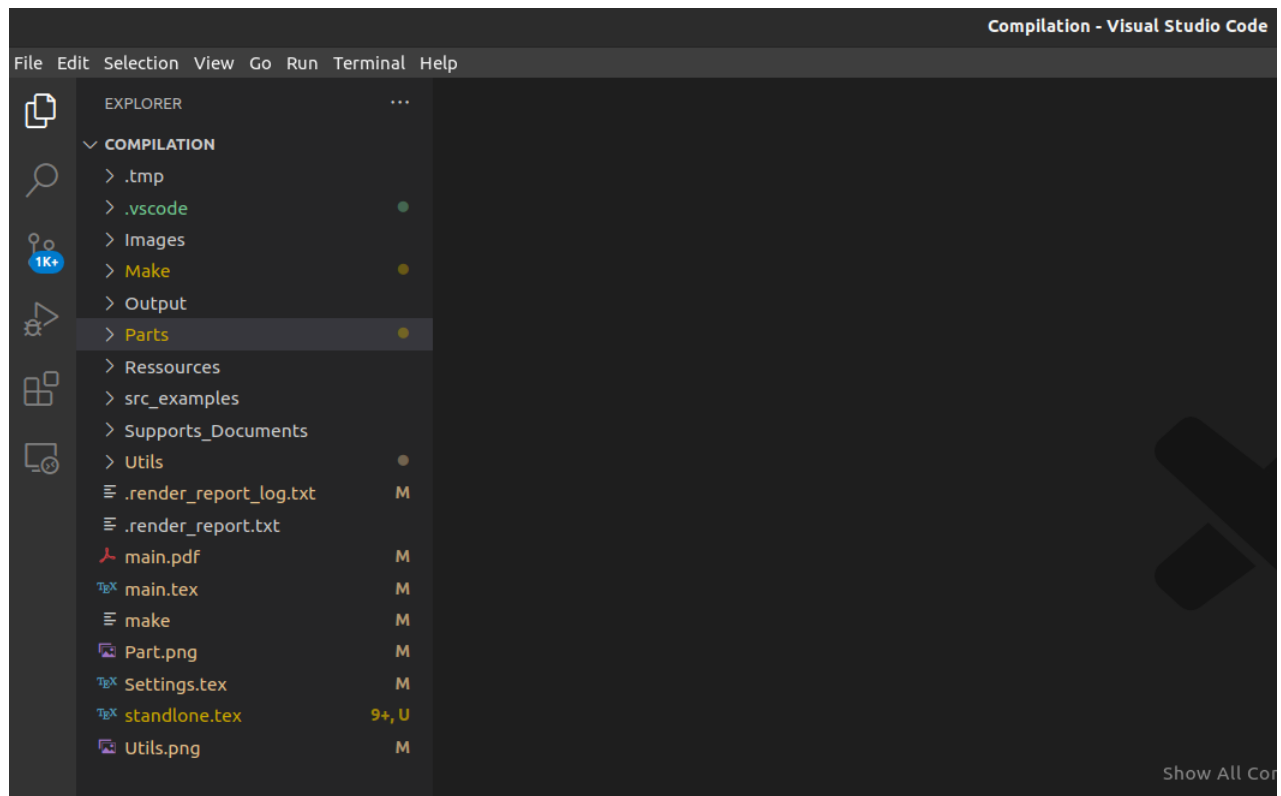


FIGURE 5.8 – Le logiciel VSCode ouvert

On observe sur la gauche l'arborescence du projet. En cliquant sur les dossiers, ces derniers se déroulent et affichent leur contenu.

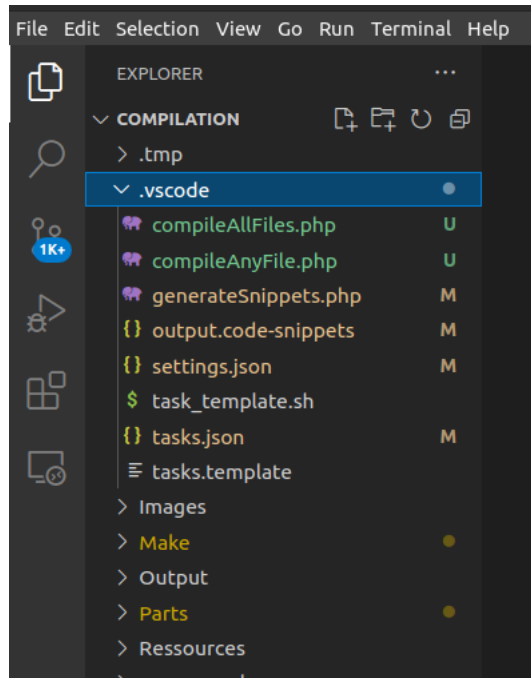


FIGURE 5.9 – L’arborescence du projet

Lancement de la compilation

Il y a deux méthodes pour lancer les outils de compilation sous VSCode :

- Via les raccourcis clavier (Tâches)
- Via les boutons sur l’interface graphique

Les raccourcis clavier

L’ensemble des commandes et outils sont disponibles en saisissant le raccourci **KEY** `CTRL+SHIFT+B`

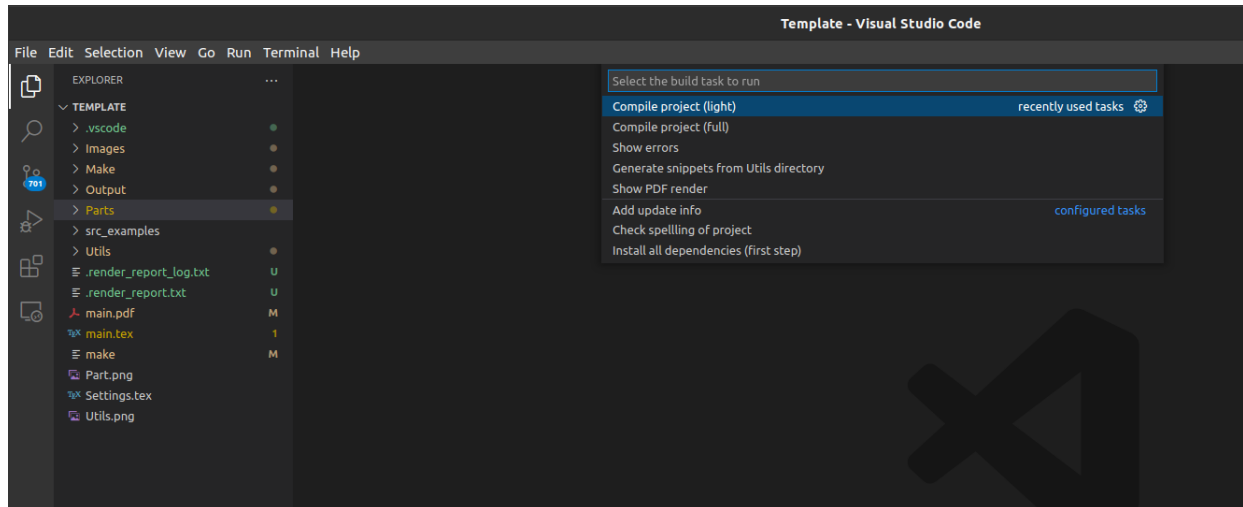


FIGURE 5.10 – Visualisation des commandes

Voici les commandes possibles :

- ▶ Compilation du projet (complète)
- ▶ Afficher le rendu PDF
- ▶ Compilation du projet sans bibliographie, sans sommaire et sans nomenclature (compilation légère)
- ▶ Génération des snippets VSCode depuis la bibliothèque Utils (auto-complétion)
- ▶ Autoriser la compilation complète du répertoire
- ▶ Vérification orthographique des fichiers
- ▶ Ajout d'une version de projet
- ▶ Retirer tous les documents de compilation
- ▶ Affichage des erreurs
- ▶ Installation des utilitaires de compilation

Les boutons

Au lieu d'utiliser les raccourcis avec KEY CTR+SHIFT+B, il est possible d'exécuter les outils avec les boutons en bas de VSCode :

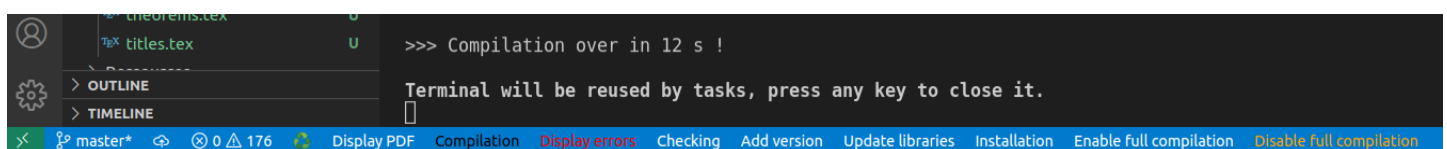


FIGURE 5.11 – Visualisation des boutons

Il faut pour cela installer le plugin **VsCode Action Buttons** .
Pour installer un plugin, il faut cliquer sur le bouton ci-dessous :

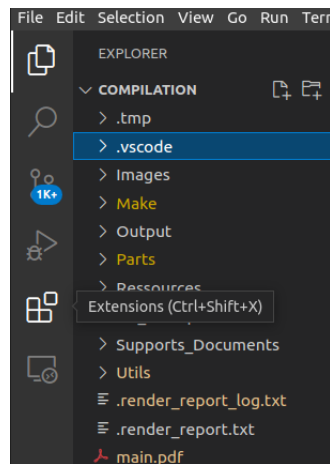


FIGURE 5.12 – Installations des plugins

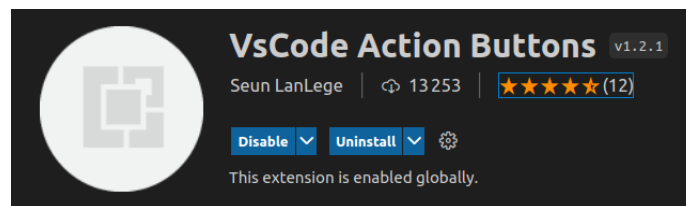


FIGURE 5.13 – Nom de l'extension

SECTION 6

MISE À JOUR DES BIBLIOTHÈQUES

Paramétrage du Git distant

Il convient tout d'abord de préciser l'url du Git distant.

Cela se fait en saisissant la commande suivante :

```
./make --change-git-folder
```

Mise à jour de l'adresse Git

ou bien en cliquant sur le bouton 

A ce moment là, on vous précise l'URL actuelle et on vous demande la future adresse :

```
Changing Git folderActual remote folder : https://github.com/nicolasleguerroue/Utils-latex.git
New folder url : 
```

FIGURE 6.1 – La nouvelle URL du Git distant

Cette adresse doit impérativement pointer sur un répertoire appelé **Utils** et qui contient les fichiers des bibliothèques au format **.sty**

Ensuite, deux étapes s'offrent à vous :

- ▶ Mettre à jour vos bibliothèques locales en récupérant celles sur le répertoire Git (pull)
- ▶ Sauvegarder vos bibliothèques locales sur le répertoire Git (push)

Architecture des mises à jours

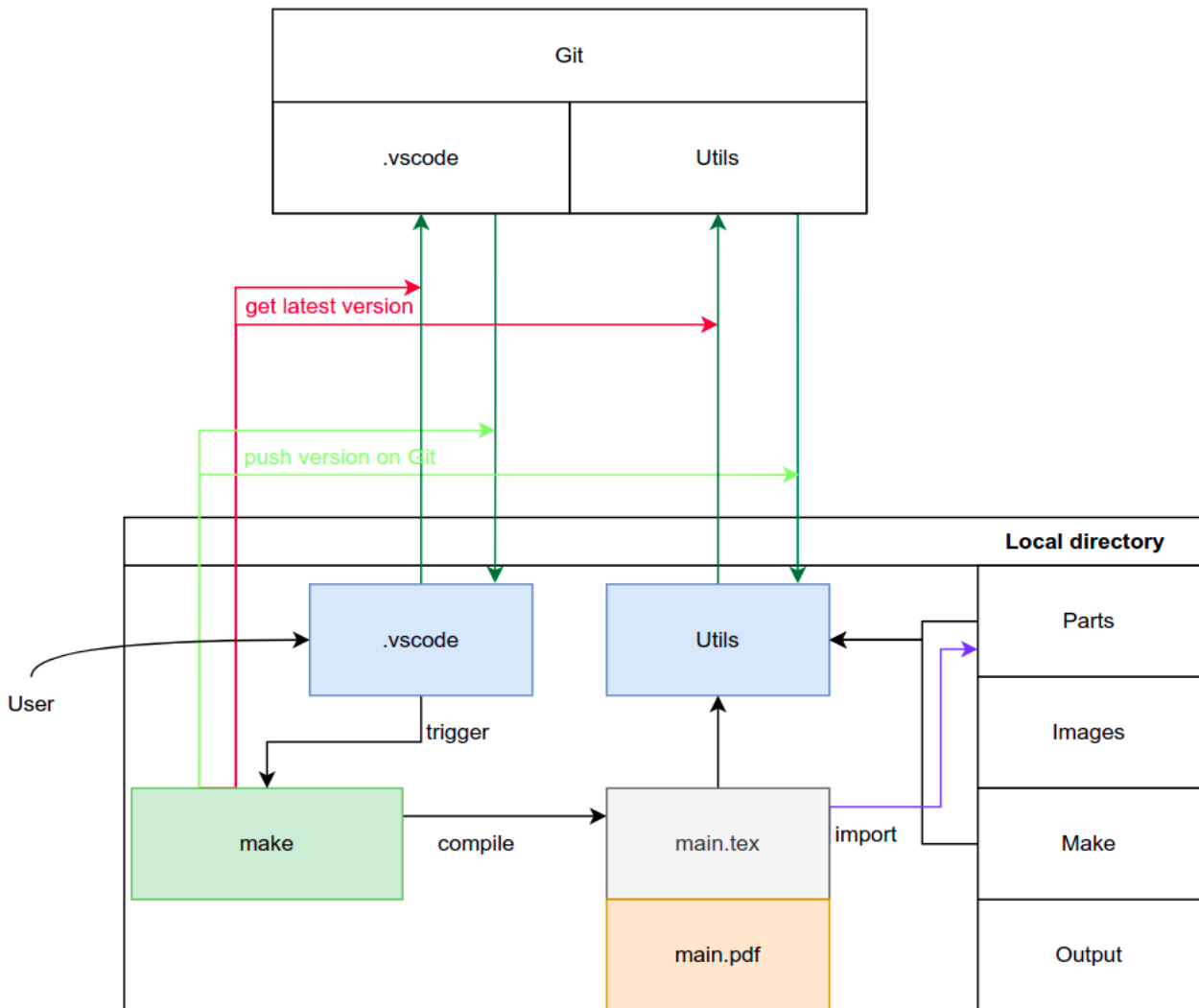



FIGURE 6.2 – Architecture des mises à jour

Dans les deux cas, le fichier  `make` va lancer les commandes pour synchroniser les bibliothèques.

Mise à jour des bibliothèques locales

Si on souhaite mettre à jour les bibliothèques locales :

```
./make --update-pull
```

Mise à jour Git

ou bien en cliquant sur le bouton  `Update local libraries`

Mise à jour des bibliothèques distantes

Si on souhaite mettre à jour les bibliothèques distantes :

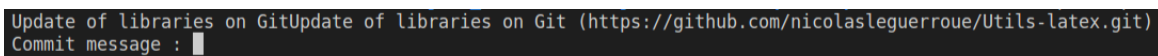
```
./make --update-push
```

Mise à jour Git

ou bien en cliquant sur le bouton 

On vous demandera la première fois votre mot de passe de compte Git pour associer VSCode à Git.

Ensuite, on vous demande un message de **commit** :



```
Update of libraries on GitUpdate of libraries on Git (https://github.com/nicolasleguerroue/Utils-latex.git)
Commit message : █
```

FIGURE 6.3 – Message de commit

SECTION 7

CRÉATION D'UN NOUVEAU PROJET

Pour créer un nouveau projet, il suffit de copier le fichier  et de le mettre là où on souhaite créer le nouveau projet.

Remarque

Il faut auparavant que le dossier **.utils_lib** soit situé à la racine de votre espace personnel (\$HOME)

```
./make --init
```

Nouveau projet

SECTION 8

FUSION DE PROJETS

Le choix d'un dossier par partie (Parts/XXX) permet de fusionner très facilement des projets.

Pour fusionner deux projets, il suffit de copier-coller le contenu du dossier `DIR Images` et `DIR Parts` du projet A dans le dossier de projet qui contiendra la fusion (projet B). Lors de la compilation, `LIB make` va gérer la fusion automatiquement.

Conventions

`LOC Header` veut dire que le code est à mettre avant `begin{document}`

`LOC Body` veut dire que le code est à mettre entre `begin{document}` et `end{document}`

`LOC main.tex` veut dire que le code est à mettre dans le fichier `FILE main.tex`

`LOC Make/Nomenclature.tex` veut dire que le code est à mettre dans le fichier `FILE Nomenclature.tex` situé dans le dossier `DIR Make`

`LOC Make/Bibliography.bib` veut dire que le code est à mettre dans le fichier `FILE Bibliography.bib` situé dans le dossier `DIR Make`

SECTION 9

BIBLIOTHÈQUE BADGES

Création de badges avec des couleurs

ElectroniqueMécaniqueInformatiqueBody

```
\badge{white}{black}{Electronique}  
\badge{white}{blue}{Mécanique}  
\badge{white}{green}{Informatique}
```

Code pour la création de badges avec des couleurs

SECTION 10

BIBLIOTHÈQUE BIBLIOGRAPHY

Format de la bibliographie

La bibliographie contenue dans le fichier  **Make** /  **Bibliography.bib** doit être au format **Bib-tex** :



 **Make/Bibliography.bib**

```
@book{ID,  
  title={Titre},  
  author={Auteur},  
  year={2007},  
  note={Une description},  
  publisher={Editeur}  
},  
...
```

Format BibTex

Le format Zotero se base sur cette architecture.

Affichage de la bibliographie

Pour afficher la bibliographie située dans le dossier  **Make** /  **Bibliography.bib**, il faut saisir la commande suivante :

 **main.tex**

```
\displayBibliography{Bibliographie}{Make/Bibliography}
```

Code pour l'affichage de la bibliographie

Remarque

L'extension du fichier .bib ne doit pas être précisée !



[1] Auteur. *Titre*. Editeur, 2000. Une description.

[2] Auteur. *Titre*. Editeur, 2001. Une description.

[3] Auteur. *Titre*. Editeur, 2003. Une description.

FIGURE 10.1 – Un rendu de la bibliographie

SECTION 11

BIBLIOTHÈQUE COLORS

Afficher du contenu en couleur

Ceci est du texte en bleu
Ceci est du texte en rouge
Ceci est du texte en vert

loc Body

```
\colors{blue}{Ceci est du texte en bleu}  
\colors{red}{Ceci est du texte en rouge}  
\colors{green}{Ceci est du texte en vert}
```

Code pour l’affichage en couleur

Liste des couleurs disponibles

red

green

blue

orange

yellow

gray

brown

cyan

black

purple

magenta

rose

darkBlue

darkBrown

darkRed

darkOrange

DarkGray

SECTION 12

BIBLIOTHÈQUE ELECTRONIC

La bibliothèque **Electronic** permet de générer des chronogrammes et des schémas électriques

Création de chronogrammes fixes

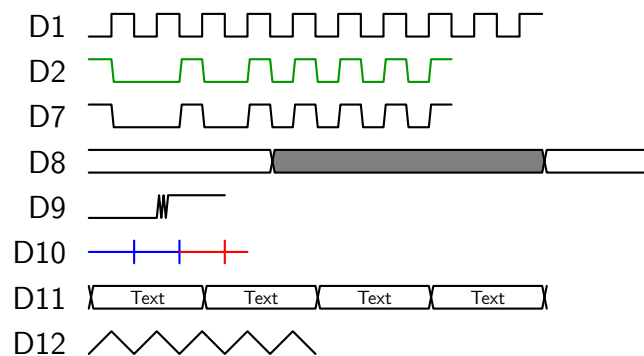


FIGURE 12.1 – Exemple 1

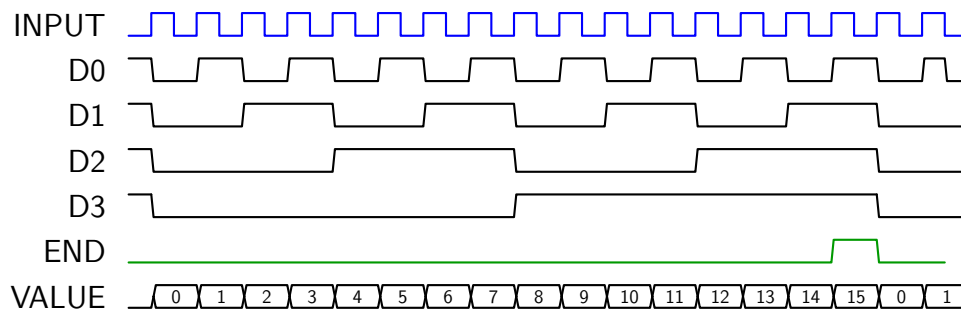


FIGURE 12.2 – Exemple 2 - Chronogramme du compteur 4 bits

Loc Body

```
\begin{numeric}{exemple 1 - chronogramme fixe}
  D1 & 20{C}  \\
  D2 & [green] 1H1L1L1H1L1L1H1L1H1L1H1L1H  \\
  D7 & [black] 1H1L1L1H1L1L1H1L1H1L1H1L1H  \\
  D8 & 8D5U7U5D  \\
```

```

D9 & LLL 2{0.1H 0.1L} 0.6H HH \\
D10 & ZZ G ZZ G XX G X \\
D11 & [d] 4{5D{Text}} 0.2D \\
D12 & [L][timing/slope=1.0] HL HL HL HL HL \\
\end{numeric}

```

Code pour la création de chronogrammes fixes [exemple 1]

loc Body


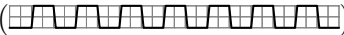
```

\begin{numeric}{Exemple 2 - Chronogramme du compteur 4 bits}
INPUT & CC [blue]16{CC} CCC \\
D0 & HL 8{LHHL} LHL \\
D1 & H 4{LLLLHHHH} LLLL \\
D2 & H 2{LLLLLLLLHHHHHHHH} LLLL \\
D3 & H{LLLLLLLLLLLLLLLLHHHHHHHHHHHHHHHHHH} LLLL \\
END & LL [green]14{LL} LHHLLL \\
VALUE & L 2D{0} 2D{1} 2D{2} 2D{3} 2D{4} 2D{5} 2D{6} 2D{7} 2D{8} 2D{9} 2D{10}
2D{11} 2D{12} 2D{13} 2D{14} 2D{15} 2D{0} 2D{1} \\
\end{numeric}%

```

Code pour la création de chronogrammes fixes [exemple 2]

Création de chronogrammes flottants

Notre signal d'horloge () provient d'un oscillateur à quartz. Notre signal d'horloge () provient d'un oscillateur à quartz.

loc Body

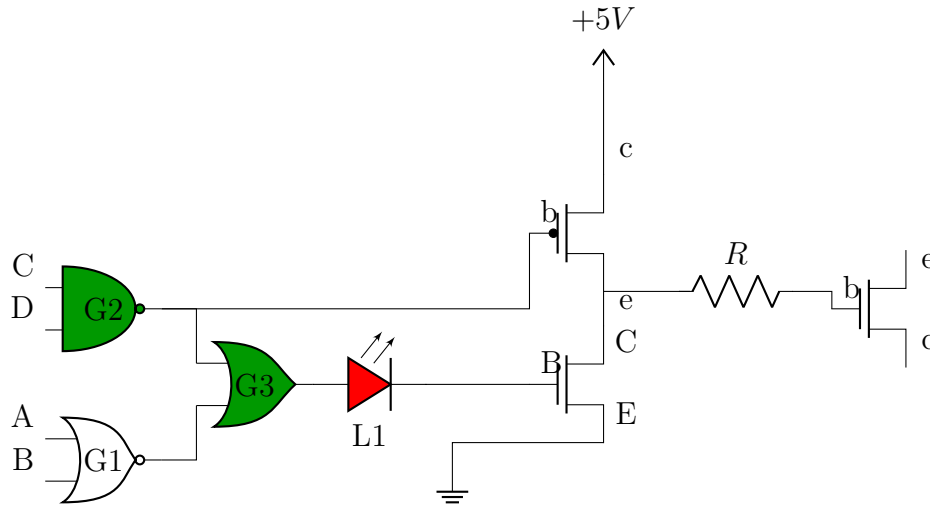
```

Notre signal d'horloge (\texttiming{[blue]CCCCC}) provient d'un oscillateur à
quartz.
Notre signal d'horloge (\texttiming[timing/draw grid]{LHLHLHLHLHLHL})
provient d'un oscillateur à quartz.

```

Code pour la création de chronogrammes flottants

Création de schémas électriques



Loc

 Body

```

\begin{schema} {Exemple de schéma électrique}

\addPower{6,5}{power1}{\$+5V\$}
\addGround{4,0}{gnd1}{\}

\setDeviceBackgroundColor{white}
\setRotate{0}
\addLogicGate{0,0}{mynor}{nor}{\}{A}{B}{G1}

\setDeviceBackgroundColor{green}
\addLogicGate{0,2}{mynand}{nand}{\}{C}{D}{G2}
\addLogicGate{2,1}{myor}{or}{\}{\}{\}{G3}
\resetColors

\addTransistor{6,1}{npnA}{nmos}{B}{C}{E}
\addTransistor{6,3}{pnpA}{pmos}{b}{e}{c}

\resetColors
\addTransistor{10,2}{npnR}{nmos}{b}{e}{c}

\addWire{mynor.out}{myor.in 2}{\orthogonalWireA}
\addWire{mynand.out}{myor.in 1}{\orthogonalWireA}

\addWire{mynand.out}{pnpA.B}{\orthogonalWireA}
\addWire{pnpA.C}{npnA.C}{\orthogonalWireA}

\addWire{pnpA.E}{power1}{\orthogonalWireA}

\addWire{npnA.E}{gnd1}{\orthogonalWireA}

```

```
\addNode{$(pnpA.C)+(1,0)}{node1}{}  
\addWire{pnpA.C}{node1}{\orthogonalWireA}  
  
\setDeviceBackgroundColor{red}  
\addLed{myor.out}{\Right}{nnpA.B}{\orthogonalWireA}{L1}  
\addResistor{node1}{\Right}{nnpR.B}{\orthogonalWireA}  
  
\end{schema}
```

Code pour la création de schémas électriques

SECTION 13

BIBLIOTHÈQUE FIGURES

Cette bibliothèque permet de créer des figures et d’afficher la liste des figures dans le document

Création d’une figure centrée

Ceci est un contenu de figure

FIGURE 13.1 – Figure en exemple

LOC Body

```
\createFigure{Figure en exemple}{Ceci est un contenu de figure}
```

Code pour la création de figure

Création d’une figure non centrée

Ceci est un contenu de figure no centrée

FIGURE 13.2 – Figure en exemple

LOC Body

```
\createNoCenteredFigure{Figure en exemple}{Ceci est un contenu de figure}
```

Code pour la création de figure non centrée

Affichage de la liste des figures

LOC main.tex

```
\displayListOfFigures{Liste des figures}
```

Code pour l’affichage de la liste des figures

SECTION 14

BIBLIOTHÈQUE FONTS

Cette bibliothèque permet d'intégrer des icons

Ajout d'icônes



SECTION 15

BIBLIOTHÈQUE GLOSSARIES

Cette bibliothèque permet de manipuler un glossaire

Ajout d'une définition au glossaire

 Make/Glossaries.tex

```
\newglossaryentry{Definiton}{name=Identifiant,description={Description}}
```

Ajout d'une définition au glossaire

Affichage du glossaire

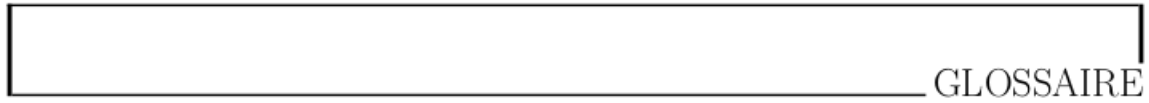
Pour afficher le glossaire située dans le dossier  Make /  Glossaries.tex, il faut saisir la commande suivante :

 main.tex

```
\displayGlossaries{Glossaire}
```

Code pour l'affichage du glossaire

Rendu du glossaire



HTML HyperText Markup Language.

I2C Inter-Integrated Circuit (Bus).

IP Internet Protocol.

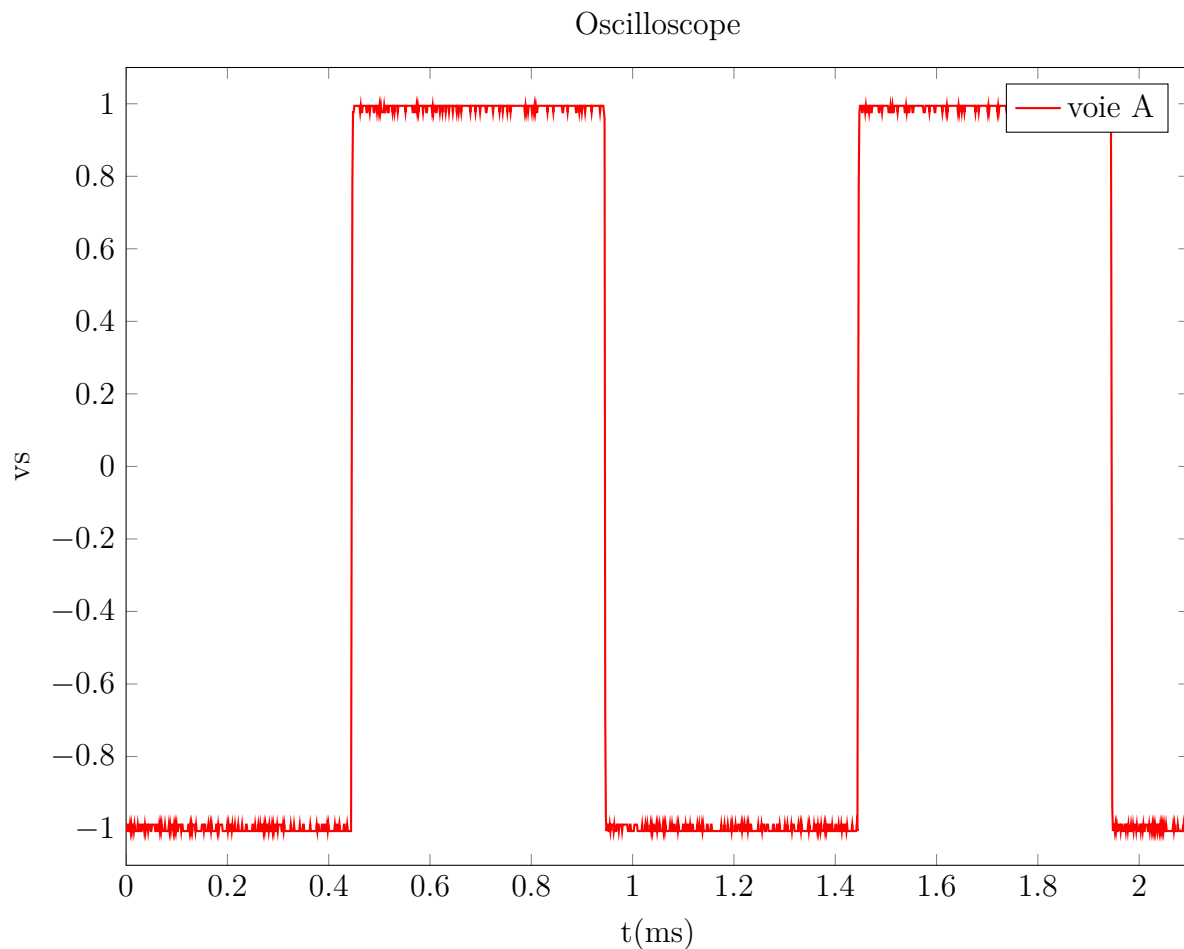
SPI Serial Peripheral Interface.

FIGURE 15.1 – Un rendu du glossaire

SECTION 16

BIBLIOTHÈQUE GRAPHIC

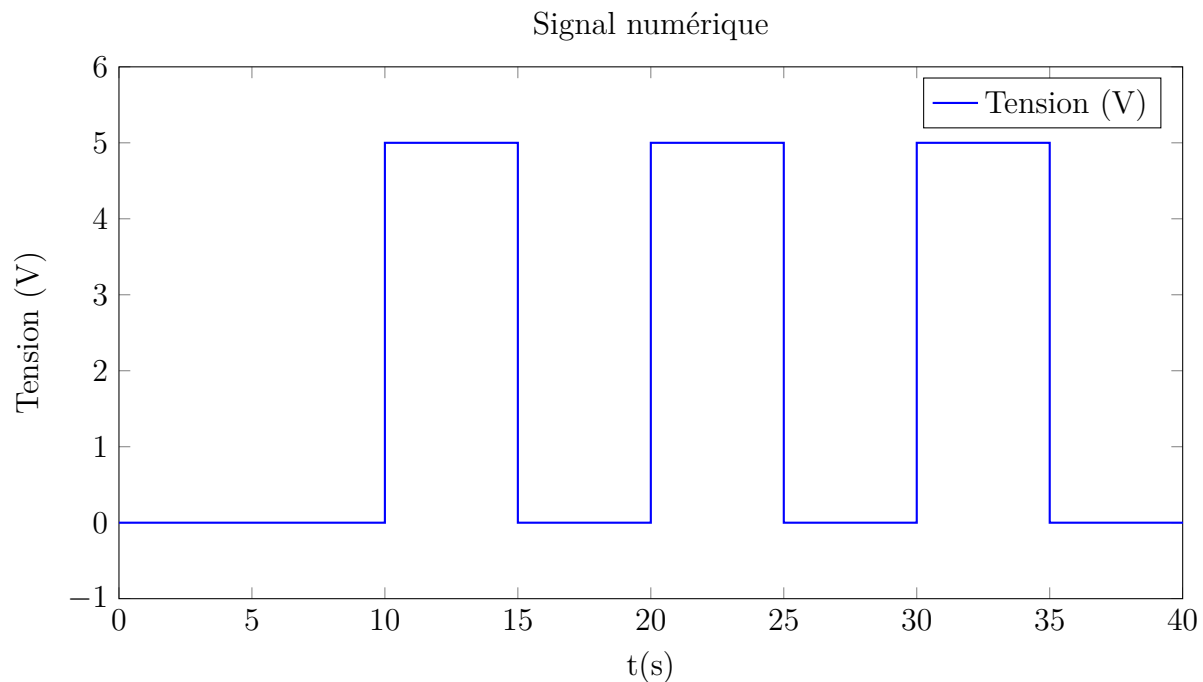
Affichage d'un graphique 2D avec insertion des données depuis un fichier txt (csv)

**loc** Body

```
\begin{graphic}{0.8}{0.6}{0}{2.1}{-1.1}{1.1}{t(ms)}{vs}{Oscilloscope}  
  \addPointsFromCSV{red}{comma}{src_examples/input_1.txt}  
  \addLegend{voie A}  
\end{graphic}
```

Code pour l’affichage d’un graphique 2D avec insertion des données depuis un fichier txt (csv)

Affichage d’un graphique 2D avec insertion des données depuis une liste de points



Loc Body

```
\begin{graphic}{0.8}{0.4}{0}{40}{-1}{6}{t(s)}{Tension (V)}{Signal numérique}
  \addPoints{blue}{(0,0)(10,0)(10,5)(15,5)(15,0)(20,0)(20,5)(25,5)(25,0)(30,0)
    (30,5)(35,5)(35,0)(40,0)}
  \addLegend{Tension (V)}
\end{graphic}
```

Code pour l’affichage d’un graphique 2D avec insertion des données depuis une liste de points

Affichage d’un graphique 2D avec insertion des données depuis une équation

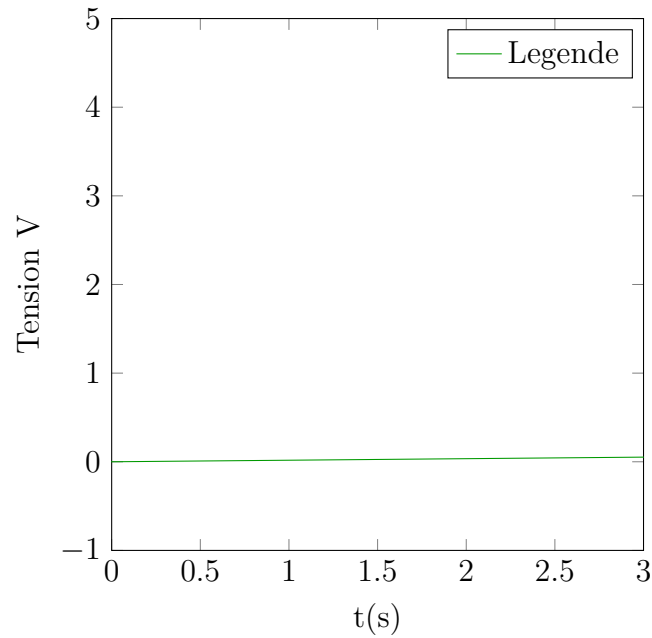


FIGURE 16.1 – Signal analogique

LOC Body

```
\begin{graphicFigure}{0.4}{0.4}{0}{3}{-1}{5}{t(s)}{Tension V}{Signal analogique}  
  \addTrace{green}{-10}{10}{sin(800*x)}  
  \addLegend{Legende}  
\end{graphicFigure}
```

Code pour l’affichage d’un graphique 2D avec insertion des données depuis une équation

Affichage de deux graphiques

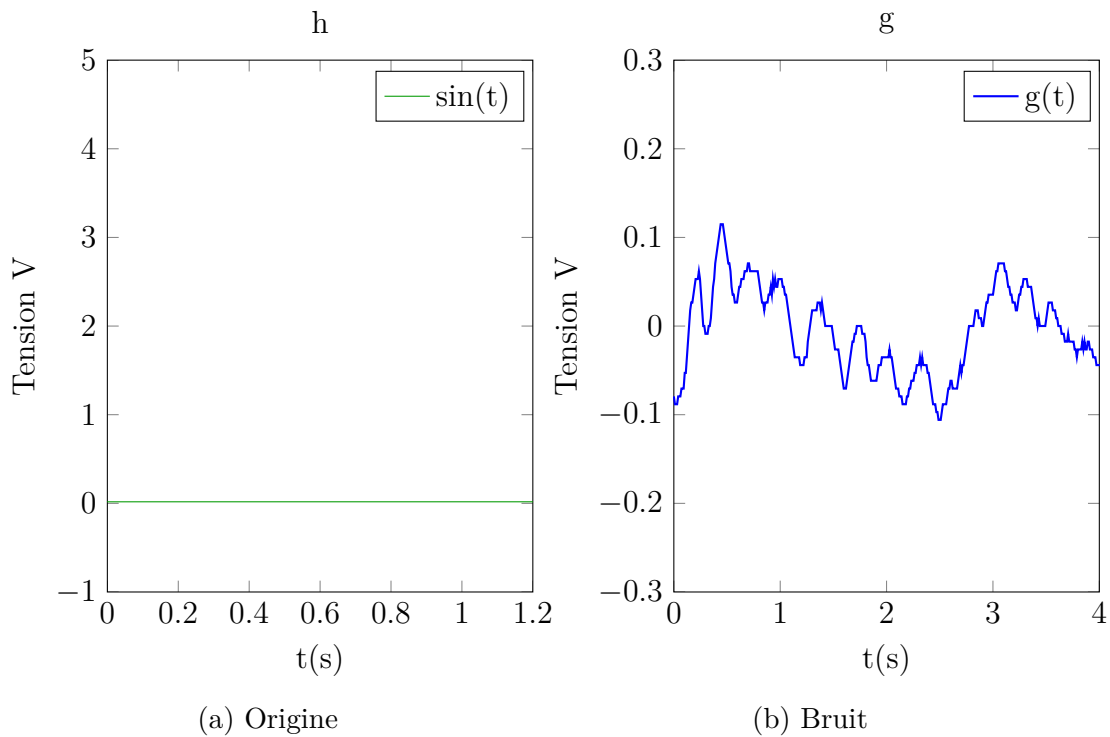


FIGURE 16.2 – Les tensions de service

loc Body

```
\begin{figure}[h!]
\centering
\begin{subfigure}[b]{0.4\linewidth}
\begin{graphic}{0.8}{1}{0}{1.2}{-1}{5}{t(s)}{Tension V}{h}
\addTrace{green}{-10}{10}{sin(x)/x}
\addLegend{sin(t)}
\end{graphic}%NO END LINE HERE
\caption{Origine}
\end{subfigure}
\begin{subfigure}[b]{0.4\linewidth}
\begin{graphic}{0.8}{1}{0}{4}{-0.3}{0.3}{t(s)}{Tension V}{g}
\addPointsFromCSV{blue}{comma}{src_examples/jack01.txt}
\addLegend{g(t)}
\end{graphic}%NO END LINE HERE
\caption{Bruit}
\end{subfigure}
\caption{Les tensions de service}
```

```
\end{figure}
```

Code pour l’affichage d’un graphique 2D avec insertion des données depuis plusieurs sources

SECTION 17

BIBLIOTHÈQUE HEADER

La bibliothèque `LIB Header` permet de gérer la mise en page globale du document, en particulier :

- ▶ Le type de page de garde :
 - ▶ Page de garde avec une image
 - ▶ Page de garde sans image
- ▶ Créer des parties avec des images (commande `LIB part` améliorée)
- ▶ Définit l'en-tête et le pied de page

Mise en forme de la page de garde avec une image

`LOC Header`

```
\setHeaderImage{Emplacement_image}{0.8}{Titre}{sous-titre}{Auteurs}{\today \\ \
  pageref{LastPage} pages}
```

Code pour la mise en forme de la page de garde avec une image

Mise en forme de la page de garde sans image

`LOC Header`

```
\setHeader{Titre}{Auteur 1 \\ Auteur 2}{Date}
```

Code pour la mise en forme de la page de garde sans image

Mise en forme de la page des parties

`LOC Body`


```
\partImg{Partie}{Images/file.png}{0.2}
```

Code pour la mise en forme de la page des parties

Ajout d'un trait entre l'en-tête et le corps de la page

loc Header

```
\setHeaderLine{0.2}
```

Code pour l'ajout d'un trait entre l'en-tête et le corps de la page

Ajout d'un trait entre le corps de la page et le bas de page

loc Header

```
\setFooterLine{0.2}
```

Code pour l'ajout d'un trait entre le corps de la page et le bas de page

Définition de la présentation globale des pages

loc Header

```
\addPresentation  
{Titre} {Centre} {\currentChapter}  
{Gauche} {} {\currentPage}
```

Code pour la définition de la présentation globale des page

Redéfinition des titres des chapitres

Par défaut, l'utilisation du mot clé chapter force à utiliser le mot-clé **Chapitre X**. Pour utiliser un autre nom, utiliser la commande suivante : **loc** Header

```
\setAliasChapter{Section}
```

Code pour la redéfinition des titres des chapitres

Mettre le document en pleine page

LOC Header

```
\setFullPage
```

Code pour mettre le document en pleine page

Récupérer le chapitre courant

Le chapitre courant sera en majuscule. **LOC** Body

```
\currentChapter
```

Code pour récupérer le chapitre courant

SECTION 18

BIBLIOTHÈQUE IMAGES

Définition de l'espace de nommage

Lors de l'ajout d'une image, il n'est pas nécessaire de connaître le nom du dossier courant.

Pour les commandes `\img`, `\imgr` et `\imgf`, l'argument **Référence** doit être de la forme **rootImages+NomImage.extension**.

La macro `\rootImages` prendra lors de la compilation la valeur du dossier courant.

Par exemple, pour ajouter une image appelée `monImage.png` provenant du dossier

`Images/Part1`, il suffit d'écrire :

```
\img{\rootImages/monImages.png}{Ma légende}{0.5}
```

Code pour l'espace de nommage

Ajout d'une image non-flottante

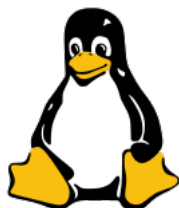


FIGURE 18.1 – Légende de l'image

`\LOC` Body

```
\img{\rootImages/tux.png}{Légende de l'image}{0.5}
```

Code pour l'ajout d'une image non-flottante

Ajout d'une image non-flottante avec une rotation

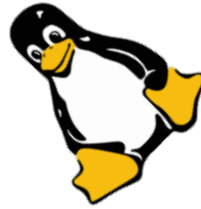


FIGURE 18.2 – Légende de l'image

LOC Body

```
\imgr{\rootImages/tux.png}{Légende de l'image}{0.5}{45}
```

Code pour l'ajout d'une image non-flottante avec une rotation

Ajout d'une image flottante

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue

SECTION 19

BIBLIOTHÈQUE ITEMS

Création d'une liste

- ▶ A
- ▶ B
- ▶ C

Loc Body

```
\begin{items}{orange}{\Triangle}  
  \item A  
  \item B  
  \item C  
\end{items}
```

Code pour la création d'une liste

Options

- ▶ Triangle
- ▶ Circle
- ▶ Bullet

```
\Triangle  
\Bullet  
\Circle
```

Options disponibles

SECTION 20

BIBLIOTHÈQUE LABELS

Création de labels générique

Il est possible d'utiliser les labels génériques :



LOC Body

```
\lbl{orange}{LIB}{Label} %Label orange
\lbl{red}{LIB}{Label} %Label rouge
\lbl{green}{LIB}{Label} %...
\lbl{magenta}{LIB}{Label}
\lbl{purple}{LIB}{Label}
\lbl{cyan}{LIB}{Label}
\lbl{blue}{LIB}{Label}
\lbl{brown}{LIB}{Label}
\lbl{yellow}{LIB}{Label}
\lbl{black}{LIB}{Label}
\lbl{rose}{LIB}{Label}
\lbl{darkBlue}{LIB}{Label}
\lbl{darkBrown}{LIB}{Label}
\lbl{darkRed}{LIB}{Label}
\lbl{darkOrange}{LIB}{Label}
\lbl{DarkGray}{LIB}{Label}
```

Code pour la création de labels génériques

Création de labels spécifiques

Contenu de fichiers

FILE Fichier
 DIR Dossier
 LINK Lien
 LIB Bibliothèque ou Logiciel
 KEY Raccourci-clavier ou touche

Réseaux

IP Adresse IP
 MAC Adresse MAC
 BIN Représentation binaire
 HEX Représentation hexadécimale

Documentation

LOC Localisation
 INFO Information périmée

Electronique

PIN Broche générique
 IN Broche d'entrée
 OUT broche de sortie
 REG Registre
 FUNC Fonction
 ADDR Adresse

LOC Body

```

\file{Fichier}
\dir{Dossier}
\link{Lien}
\lib{Bibliothèque ou Logiciel}
\shortcut{Raccourci-clavier ou touche}

\ip{Adresse IP}
\mac{Adresse MAC}
\bin{Représentation binaire}
\hexa{Représentation hexadécimale}

\loc{Localisation}
\deprecated{Information périmée}

\genericPin{Broche générique}
\inputPin{Broche d'entrée}
\outputPin{broche de sortie}
\reg{Registre}
\func{Fonction}
\adr{Adresse}
  
```

Code pour la création de labels spécifiques

SECTION 21

BIBLIOTHÈQUE LAYOUT

La mise en page est séparée en 4 parties : frontmatter (début du document, numérotation romaine), mainmatter (avant le premier chapitre), appendix (annexes) et backmatter avant les tables et bibliographies

Mise en gras

Texte en gras

LOC Body

```
\bold{Texte en gras}
```

Code pour mettre le texte en gras

Mise en italique

Texte en italique

LOC Body

```
\italic{Texte en gras}
```

Code pour mettre le texte en italique

Mise en gras et italique

Texte en gras et italique

LOC Body

```
\ib{Texte en gras et italique}
```

Code pour mettre le texte en gras et italique

Ajout d'un espace vertical

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu li-

bero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestib-

LOC Body

`\sn`

Code pour ajouter un espace vertical

SECTION 22

BIBLIOTHÈQUE LINKS

Paramétrage des liens et des méta-données

loc Header

```
%@input Titre du PDF
%@input Auteur(s)
%@input Sujet du fichier PDF (courte phrase)
%@input Créateur du fichier PDF
%@input Producteur du fichier PDF
%@input Mots-clés (liste)
%@input Couleurs des liens
%@input Couleurs des citations dans la bibliographie
%@input Couleurs des liens de fichier
\setParameters {Tutoriel Latex} {Nicolas Le Guerroué} {Bibliothèque Utils} {
  Nicolas Le Guerroué}{Latex}{green}{blue}{blue}
```

Code pour paramétrer les liens et les métadonnées

SECTION 23

BIBLIOTHÈQUE MATHS

Création d'une matrice 3*3

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

LOC

Body

```
$$\emat{a & b & c}{d & e & f}{g & h & i} $$
```

Code pour la création d'une matrice 3*3

Création d'un vecteur à trois dimensions

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

LOC

Body

```
$$\evec{a}{b}{c} $$
```

Code pour la création d'un vecteur à trois dimensions

Création d'un tenseur à trois dimensions

$$\begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}$$

 Body

```
$$\torsor{a & d}{b & e}{c & f} $$
```

Code pour la création d'un torseur à trois dimensions

SECTION 24

BIBLIOTHÈQUE MESSAGEBOX

Création de boîtes de dialogues

Message

Voici un message

Message

Et un autre message

Message

bref...

Loc Body

```
\messageBox{Message}{orange}{white}{Voici un message}{black}
```

Code pour la création de boîtes de dialogues

SECTION 25

BIBLIOTHÈQUE NOMENCLATURE

La Bibliothèque **Nomenclature** permet de générer des nomenclatures.

Ajout d'un élément dans la nomenclature




Chaque élément est de la forme

```
\nomenclature[category]{$symbole$}{définition}

%Par exemple
\nomenclature[E]{$r$}{Rapport cyclique d'un signal périodique}
```

Format pour la nomenclature

et doit être rajouté dans le fichier    Nomenclature.tex

Dans la commande, **category** indique le type de grandeur. Il y a actuellement 6 types de grandeurs définies dans le fichier    Nomenclature.sty¹ :

- P pour les **Constantes physiques**
- O pour les **Autres symboles**
- N pour les **Nombres spéciaux**
- A pour les **Amplificateurs Opérationnels**
- M pour les **Mécanique**
- E pour les **Électronique**

Ajout des unités

Pour ajouter une unité, il suffit d'invoquer la commande   à la fin du dernier argument de la commande  .

1. Cependant, rien ne vous empêche d'en ajouter en modifiant la bibliothèque

Ajout d'une unité

LOC Make/Nomenclature.tex

```
\nomenclature[A]{\varepsilon}{Tension différentielle $(\varepsilon = E_+ - E_-)$  
$\addUnit{V}}
```

Ajout d'une unité

Exemple de nomenclature

La nomenclature est dans le fichier **DIR** Make / **FILE** Nomenclature.tex.

LOC Make/Nomenclature.tex

```
\nomenclature[E]{r}{Rapport cyclique d'un signal périodique}  
\nomenclature[A]{A_d}{Coefficient d'amplification, gain différentiel }  
\nomenclature[A]{\varepsilon}{Tension différentielle $(\varepsilon = E_+ - E_-)$  
$\addUnit{V}}  
\nomenclature[A]{E_+}{Tension entrée non inverseuse \addUnit{V}}  
\nomenclature[A]{E_-}{Tension entrée inverseuse \addUnit{V}}  
\nomenclature[E]{\eta}{Rendement d'un mécanisme \addUnit{\%}}  
\nomenclature[E]{\varphi}{Déphasage entre deux signaux \addUnit{rad}}
```

Exemple de nomenclature

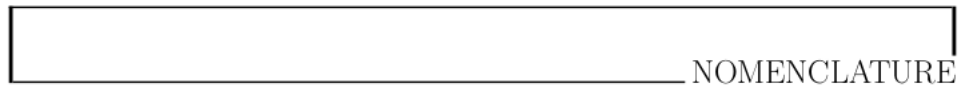
Affichage de la nomenclature

Pour afficher la nomenclature, il suffit de faire la commande :

LOC main.tex

```
\displayNomenclature{Nomenclature}{Make/Nomenclature.tex}
```

Code pour l'affichage de la nomenclature

**Amplificateurs Opérationnels**

ε	Tension différentielle ($\varepsilon = E_+ - E_-$)	V
A_d	Coefficient d'amplification, gain différentiel	
E_+	Tension entrée non inverseuse	V
E_-	Tension entrée inverseuse	V

Électronique

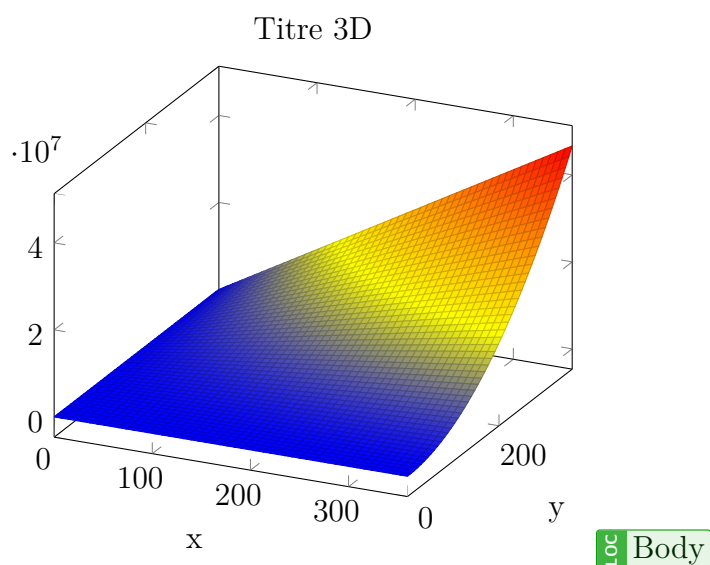
η	Rendement d'un système	%
φ	Déphasage entre deux signaux	rad
r	Rapport cyclique d'un signal périodique	

FIGURE 25.1 – Rendu d'une nomenclature

SECTION 26

BIBLIOTHÈQUE OBJECT3D

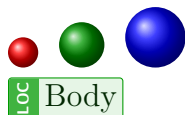
Affichage d'un graphique 3D avec insertion des données depuis une équation



```
\plot{Titre 3D}{x*y*y}
```

Code pour l'affichage graphique 3D avec insertion des données depuis une équation

Affichage de sphères en 3D



```
\ball{red}{2}  
\ball{green}{3}  
\ball{blue}{4}
```

Code pour l'affichage de sphères en 3D

SECTION 27

BIBLIOTHÈQUE PDF

Insertion d'un document PDF

loc Body

```
\includepdf [page=1,2,3]
```

Code pour ajouter un document au format PDF

Insertion d'un ensemble de pages d'un document PDF

loc Body

```
\includepdf [page=1,2,3]
```

Code pour ajouter un ensemble de page d'un document au format PDF

SECTION 28

BIBLIOTHÈQUE PROGRAMMING

Affichage d'un code C/C++ avec titre

```
#include <iostream>

#define CONST 1

int var = 1;
float

int main() {

    call();
    return 0;

} //End main
```

Titre

loc Body

```
\begin{Cpp}{Titre}
    #include <iostream>

    #define CONST 1

    int var = 1;
    float

    int main() {

        call();
        return 0;

    } //End main
```

```
\end{Cpp}
```

Code pour l’affichage d’un code C/C++ avec titre

Affichage d’un code C/C++ sans titre

```
#define CONST 1 #const var

int var = 1;
float g = 2.5;
...
```

loc Body

```
\begin{Cpp}
#include <iostream>

#define CONST 1 #const var

int var = 1;
float g = 2.5;
...

\end{Cpp}
```

Code pour l’affichage d’un code C/C++ sans titre

Affichage d’un code Python avec titre

```
def call(input):

    """docstring"""
    a = input
    for elem in a:
        print(elem) #show
```

Titre du code

loc Body

```
\begin{Python}{Titre du code}
def call(input):

    """docstring"""
    a = input
```

```
for elem in a:
    print(elem) #show
\end{Python}
```

Code pour l’affichage d’un code Python avec titre

Affichage d’un code Python sans titre

```
"""docstring"""
...
```

LOC Body

```
\begin{Python}
def call(input):

    """docstring"""
    ...
\end{Python}
```

Code pour l’affichage d’un code Python sans titre

Affichage d’un code Bash avec titre

```
sudo apt-get -y update
sudo apt-get -y upgrade
echo -e "content"
```

Titre du code

LOC Body

```
\begin{Bash}{Titre du code}
sudo apt-get -y update
sudo apt-get -y upgrade
echo -e "content"
\end{Bash}
```

Code pour l’affichage d’un code Bash avec titre

Affichage d’un code bash sans titre

Ce type d’affichage n’est pas encore supporté par la bibliothèque.

SECTION 29

BIBLIOTHÈQUE QUOTES

Création d'une citation

Il est possible de créer des citations uniquement dans la classe **book** , **report** et **utils_report**

« *Ceci est une citation* »

Nicolas LE GUERROUE

 Body

```
\addQuote{Nicolas LE GUERROUE}{Ceci est une citation}
```

Code pour la création d'une citation

SECTION 30

BIBLIOTHÈQUE TABLES

Création d'un tableau

U_A (V)	U_B (V)	Sens du courant	$U_A - U_B$
10	5	De A vers B	5
5	10	de B vers A	-5
5	5	Aucun courant ne circule	0

FIGURE 30.1 – Réponse sur le sens du courant en fonction des tensions U_A et U_B

LOC

Body

```

\begin{figure}[!h]
  \centering
  \begin{tabular}{|c|c|c|c|}
    \hline
    $U_A$ (V) & $U_B$ (V) & Sens du courant & $U_A-U_B$\\
    \hline
    10 & 5 & \color{blue}{De A vers B} & 5\\
    \hline
    5 & 10 & \color{blue}{de B vers A} & -5\\
    \hline
    5 & 5 & \color{blue}{Aucun courant ne circule} & 0\\
    \hline
  \end{tabular}
  \caption{Réponse sur le sens du courant en fonction des tensions $U_A$ et $U_B$}
\end{figure}

```

Code d'exemple

Affichage de la liste des tables

loc main.tex

```
\displayListOfTables{Liste des tables}
```

Code pour l’affichage de la liste des tables

L’argument passé à la macro est le nom visible sur la page ainsi que dans la table des matières.

SECTION 31

BIBLIOTHÈQUE THEOREMS

Les Theorèmes sous Latex sont des sections d'informations précises (remarque, question, réponse, propriété...) mais dont la mise en page es transparente tout comme la numérotation.

Par exemple, on souhaite créer une question :

Création d'une question

Question 1. *Quelle heure est-il ?*

LOC Body

```
\begin{question}
  Quelle heure est-il ?
\end{question}
```

Code pour la création d'une question

L'ajout d'une deuxième question se fait en ajoutant le même code.

Question 2. *Quelle est ma deuxième question ?*

Création d'une reponse

>>> **1.** *il est 18 h.*

LOC Body

```
\begin{reponse}
  il est 18 h.
\end{reponse}
```

Code pour la création d'une reponse

Création d'une propriete

Propriété 1. *Un produit scalaire est commutatif.*

LOC Body

```
\begin{propriete}
  Un produit scalaire est commutatif.
\end{propriete}
```

Code pour la création d'une propriete

Création d'une proposition

Proposition 1. *Les chats sont des mammifères.*

LOC Body

```
\begin{proposition}
  Les chats sont des mammifères.
\end{proposition}
```

Code pour la création d'une proposition

Création d'une remarque

Remarque 1. *remarque sur Latex*

LOC Body

```
\begin{remarque}
remarque sur Latex
\end{remarque}
```

Code pour la création d'une remarque

Création d'un exemple

Exemple 1. *Ceci est un exemple d'exemple*

LOC Body

```
\begin{exemple}  
  Ceci est un exemple d'exemple  
\end{exemple}
```

Code pour la création d'une exemple

Création d'une définition

Definition 1. *Une phrase est un ensemble de mots.*

 Body

```
\begin{definition}  
  Une phrase est un ensemble de mots.  
\end{definition}
```

Code pour la création d'une definition

Création d'une solution

Solution 1. *La solution est triviale*

 Body

```
\begin{solution}  
  La solution est triviale  
\end{solution}
```

Code pour la création d'une solution

SECTION 32

BIBLIOTHÈQUE TITLES

Titre de chapitre

LOC Body

```
\chapter{Titre}
```

Code pour l'ajout d'un titre

Titre de section

LOC Body

```
\section{Titre de section}
```

Code pour l'ajout d'une section

Titre de sous-section

LOC Body

```
\subsection{Titre de sous-section}
```

Code pour l'ajout d'une sous-section

Titre de sous-sous-section

LOC Body

```
\subsection{Titre de sous-sous-section}
```

Code pour l'ajout d'une sous-sous-section

SECTION 33

BIBLIOTHÈQUE TREE

La bibliothèque  **Tree** permet de générer des arborescences de dossiers et fichiers.

Création d'une arborescence simple


```
Root1
├── SubRoot1
│   ├── SubSubRoot1
│   ├── SubSubRoot2
│   └── SubSubRoot3
├── SubRoot2
└── SubRoot3
```

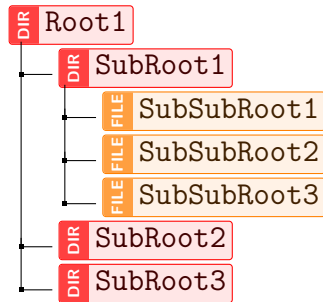
```
\dirtree{%
.1 Root1.
.2 SubRoot1.
.3 SubSubRoot1.
.3 SubSubRoot2.
.3 SubSubRoot3.
.2 SubRoot2.
.2 SubRoot3.
}
```

Code pour la création d'un arbre simple

Le **.Numéro** représente la profondeur de l'arborescence, c'est à dire le niveau. Chaque fin de ligne doit impérativement se terminer par un point et la ligne **dirtree{** doit se terminer par un symbole '%'.

Création d'une arborescence plus évoluée

Avec l'utilisation de la bibliothèque  **Labels**, il est possible de mettre des couleurs très facilement.



```

\dirtree{%
.1 \dir{Root1}.
.2 \dir{SubRoot1}.
.3 \file{SubSubRoot1}.
.3 \file{SubSubRoot2}.
.3 \file{SubSubRoot3}.
.2 \dir{SubRoot2}.
.2 \dir{SubRoot3}.
}

```

Code pour la création d'une arborescence plus évoluée

Création d'une arborescence dans une figure

Avec la bibliothèque **LIB Figures**, on peut intégrer une arborescence au sein d'une figure non centrée.

Actuellement, on ne peut pas centrer l'arbre sous peine de dégrader le rendu.

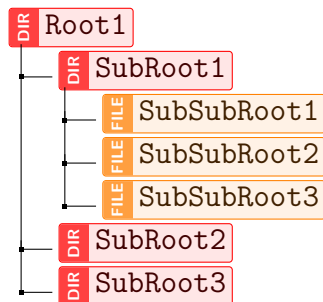


FIGURE 33.1 – Exemple d'arborescence

```

\createNoCenteredFigure{Exemple d'arborescence}{
  \dirtree{%
    .1 \dir{Root1}.
    .2 \dir{SubRoot1}.
    .3 \file{SubSubRoot1}.
    .3 \file{SubSubRoot2}.
    .3 \file{SubSubRoot3}.
  }
}

```

```
.2 \dir{SubRoot2}.  
.2 \dir{SubRoot3}.  
}  
  
}
```

Code pour la création d'une arborescence dans une figure

LISTE DES FIGURES

4.1	Arborescence du projet	12
5.1	Nombre de ligne pour les parties	16
5.2	Nombre de ligne pour les bibliothèques	17
5.3	Message d'ajout d'éléments de la bibliothèque Utils	17
5.4	Message d'avertissements	18
5.5	Vérification orthographique	18
5.6	Commande de vérification orthographique	18
5.7	Ajout d'une version	19
5.8	Le logiciel VSCode ouvert	20
5.9	L'arborescence du projet	21
5.10	Visualisation des commandes	22
5.11	Visualisation des boutons	22
5.12	Installations des plugins	23
5.13	Nom de l'extension	23
6.1	La nouvelle URL du Git distant	24
6.2	Architecture des mises à jour	25
6.3	Message de commit	26
10.1	Un rendu de la bibliographie	31
12.1	Exemple 1	34
12.2	Exemple 2 - Chronogramme du compteur 4 bits	34
13.1	Figure en exemple	38
13.2	Figure en exemple	38
15.1	Un rendu du glossaire	42
16.1	Signal analogique	45
16.2	Les tensions de service	46

18.1 Légende de l'image	51
18.2 Légende de l'image	52
25.1 Rendu d'une nomenclature	64
30.1 Réponse sur le sens du courant en fonction des tensions U_A et U_B	71
33.1 Exemple d'arborescence	78