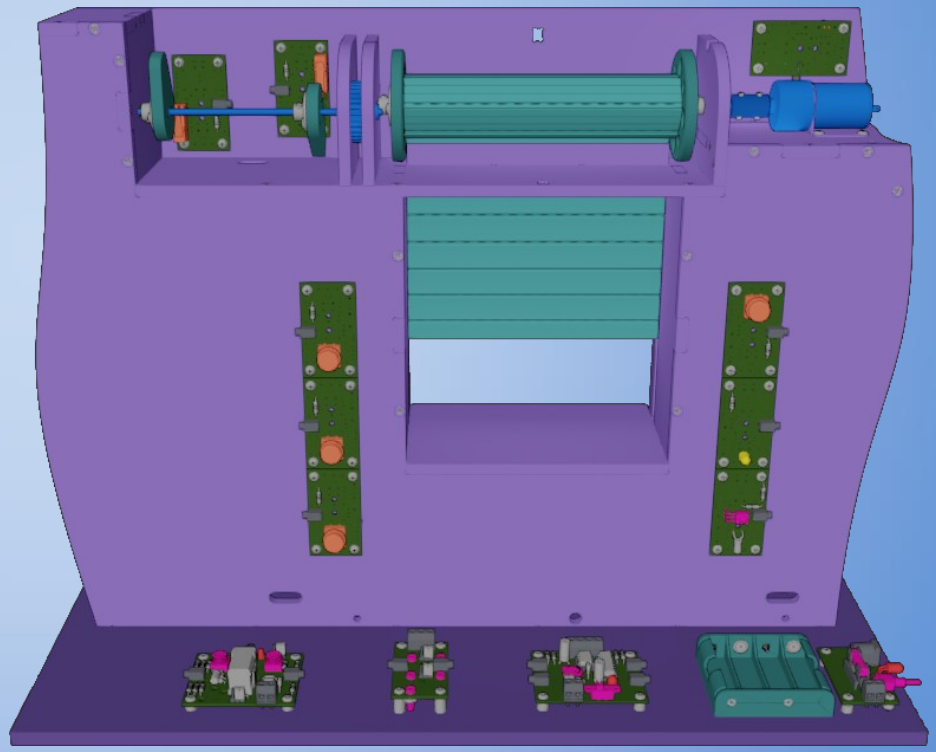
Séquence 3

Algorithme et programmation

SI
LYCÉE LÉONARD DE VINCI

Document Technique

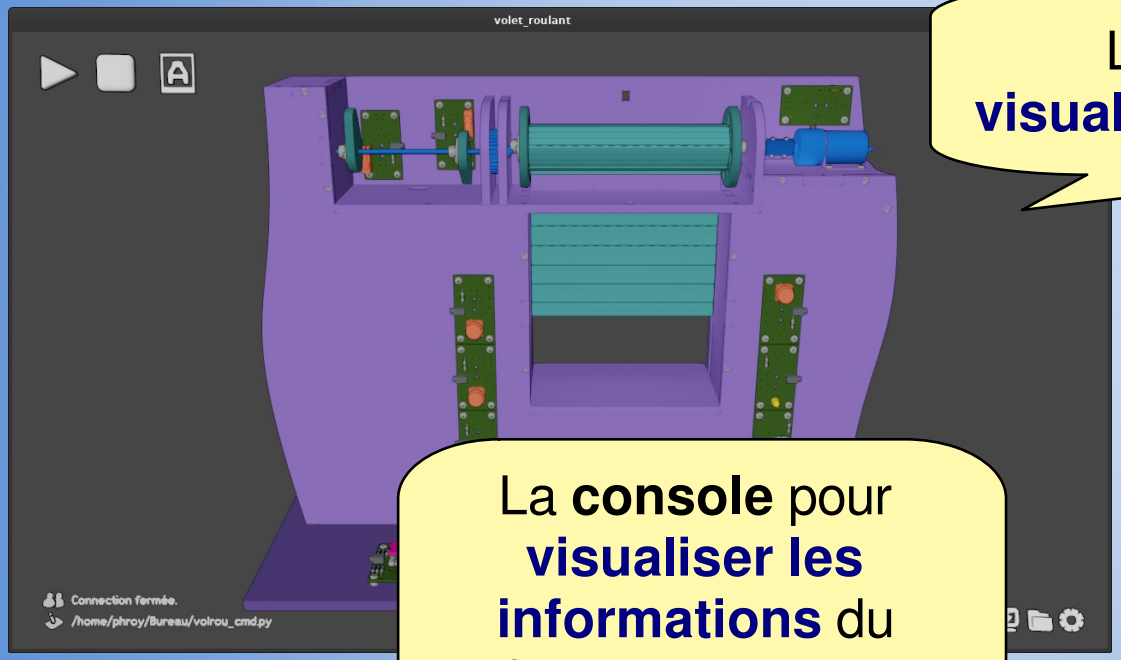
Jumeau numérique d'un volet roulant



https://forge.aeif.fr/blender-edutech/jumeau-numerique

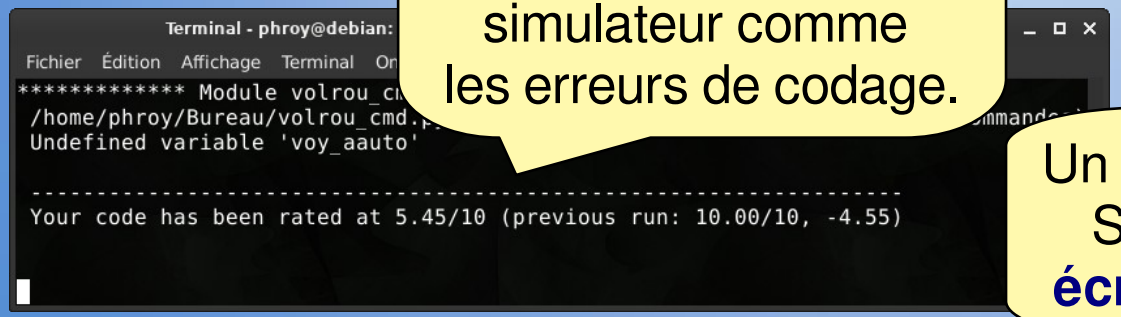
Présentation du jumeau numérique et de son environnement de programmation

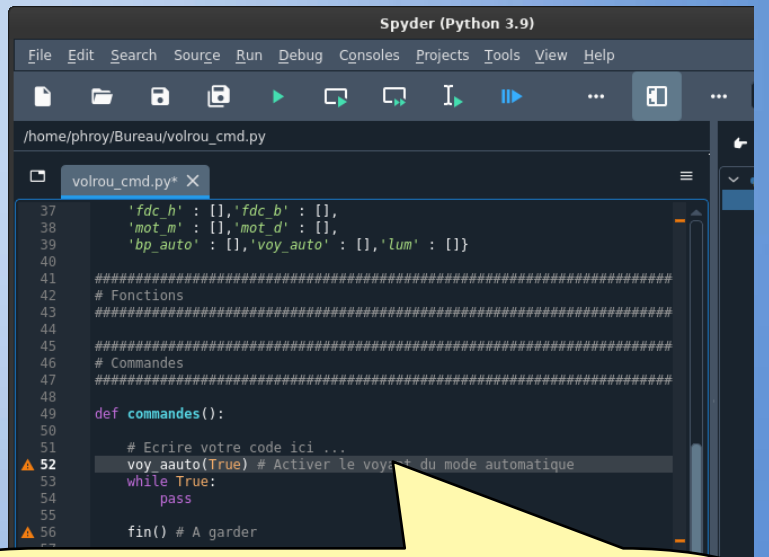
Le jumeau numérique est une maquette numérique qui se commande grâce au langage **Python**. L'interface de programmation se décompose en **3 fenêtres** : un éditeur de texte, le simulateur et la console.



Le **simulateur** permet de **visualiser l'évolution du système.**

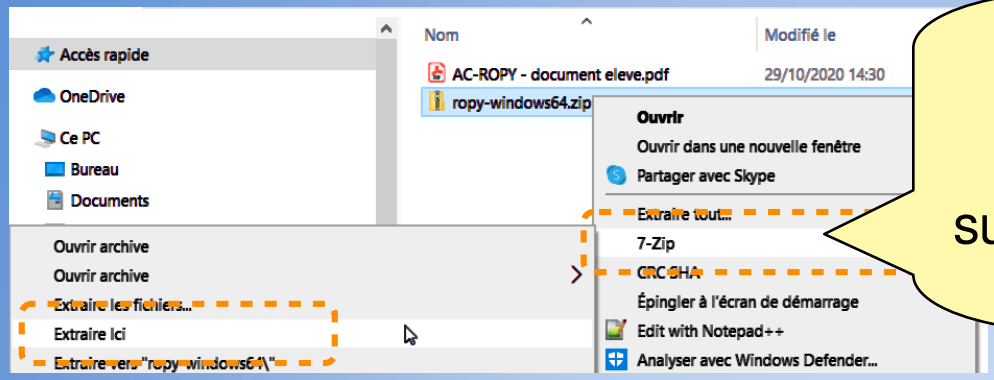
La **console** pour **visualiser les informations** du simulateur comme les erreurs de codage.





Un **éditeur de texte** (Notepad++, Spyder, Atom, Emacs, …) pour **écrire le programme** en **Python**.

Mettre en place l'environnement de développement



1 : Récupérer l'archive **volet_roulant-windows64.zip** et la décompresser avec **7-Zip** sur le **bureau**. L'extraction va créer le répertoire volet_roulant.

Le **simulateur** et la **console** se lancent en même temps avec **volet_roulant.bat** (situé dans le répertoire créé).

Exécuter le programme

Afficher l'**aide**

Arrêter et **réinitialiser**

Message avec le **jumeau réel**

Fichier de commandes

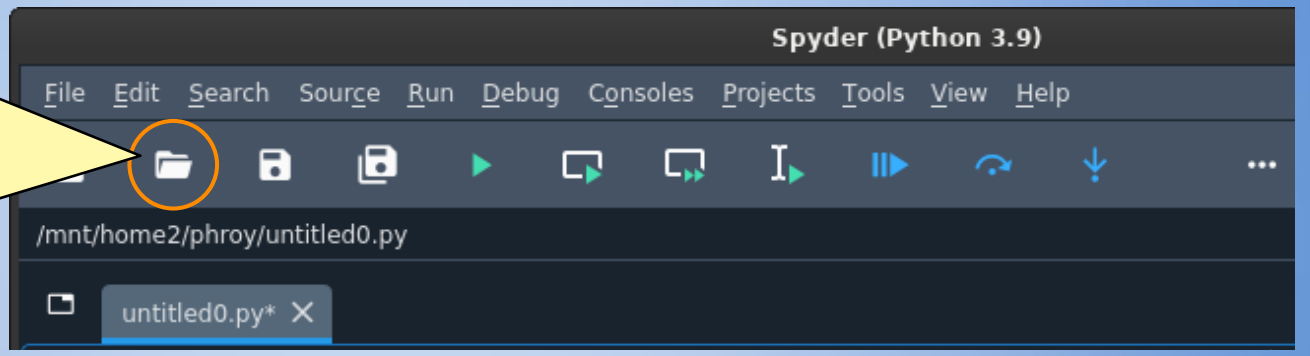
Maquette numérique du système en fonctionnement

Mettre en place l'environnement de développement

2 : Copier **dans votre répertoire** le fichier de commandes : **volrou_cmd.py** (volet roulant commandes).

3 : Lancer **Spyder**.

4 : Dans **Spyder** **ouvrir** le fichier de commandes qui a été précédemment copié dans votre répertoire.



5 : Dans le **simulateur**, **définir** votre fichier comme fichier de commandes.

6 : Le nom de votre fichier doit apparaître ici.

Mettre en place l'environnement de développement

8 : **Sauvegarder** le fichier

Attention !

Toujours sauvegarder le fichier avant son exécution.

9 : **Exécuter** le programme.

7 : **Écrire** le code Python.

10 : Si votre code a **une erreur**, la console indique où elle se trouve. Ici la ligne 52 contient une commande indéfinie : `'voy_aauto'`.

```
File Edit Search Source Run Deb
/home/phroy/Bureau/volrou_cmd.py
volrou_cmd.py
commandes
volet_roulant
Terminal - phroy@debian: /mnt/home2/phroy/Bureau/volet_r
Fichier Édition Affichage Terminal Onglets Aide
```

```
32  ##########################################
33
34  # Brochage du volet roulant
35  brochage={
36      'bp_m' : [],'bp_a' : [],'bp_d' : [],
37      'fdc_h' : [],'fdc_b' : [],
38      'mot_m' : [],'mot_d' : [],
39      'bp_auto' : [],'voy_auto' : [],'lum' : []}
40
41  ##########################################
42  # Fonctions
43  ##########
44
45  ##########
46  # Commandes
47  ##########
48
49  def commandes():
50
51      # Ecrire votre code ici ...
52      voy_aauto(True) # Activer le voyant du mode
53      while True:
54          pass
55
56      fin() # A garder
57
58
59  ##########################################
60  # En: External call << DONT CHANGE THIS SECTION
61  # Fr: Appel externe << NE PAS MODIFIER CETTE SEC
62  ##########################################
```

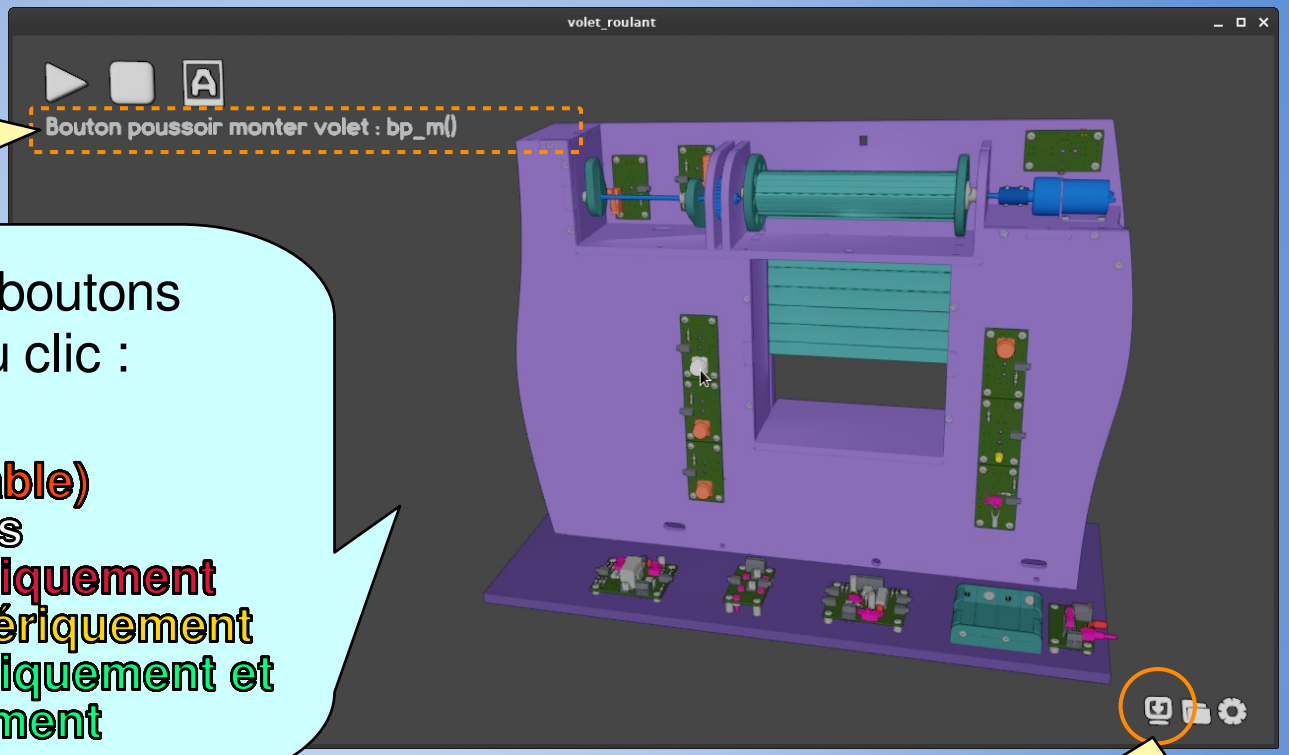
```
conda: base (Python 3.9.13)   Completions: conda
```

```
************* Module volrou_cmd
/home/phroy/Bureau/volrou_cmd.py:52: error (E0602, undefined-variable, commandes)
Undefined variable 'voy_aauto'

-------------------------------------------------------------------
Your code has been rated at 5.45/10 (previous run: 10.00/10, -4.55)
```

Manipulation de la maquette numérique

Description du composant qui a le focus de la souris



Les capteurs et les boutons sont sensibles au clic :

- **Magenta :** passif
- **Orange :** actif (activable)
- **Blanc :** focus souris
- **Rouge :** activé physiquement
- **Jaune :** activé numériquement
- **Vert :** activé physiquement et numériquement

Réinitialisation de la vue

Le **bouton du centre** sert à **manipuler** le modèle 3D :

- **Clic centre** : Rotation du mécanisme (Orbit)
- **Clic centre + Maj** : Déplacement du mécanisme (Pan)
- **Clic centre + Ctrl** : Zoom
- **Molette** : Zoom

Acquisition de données

Il possible de suivre les valeurs des entrées/sorties ainsi que des grandeurs physiques du systèmes (position, vitesse).

Dans le script Python, l'**enregistrement des données est activé** par la commande `daq([variables])`. `'[variables]'` est la liste des variables à suivre. Un fichier de données au format CSV sera généré à la fin du cycle. Par exemple : `daq(['bp_m', 'bp_d', 'mot_angle']}`.

L'affichage **du graphique** est déclenchée par `plot([variables])`. `'[variables]'` est la liste des variables à visualiser (variables enregistrées avec la commande `daq`). Par exemple : `plot(['bp_m','mot_angle']}`.

Données : `'bp_m'`, `'bp_m_r'`, `'bp_a'`, `'bp_a_r'`, `'bp_d'`, `'bp_d_r'`, `'mot_m'`, `'mot_d'`, `'fdc_h'`, `'fdc_h_r'`, `'fdc_b'`, `'fdc_b_r'`, `'bp_auto'`, `'bp_auto_r'`, `'voy_auto'`, `'lum'`, `'lum_r'`, `'t' (temps)`, `'mot_angle'` et `'mot_vitesse'`.

`*_r` correspond à la valeur de la variable du jumeau réel.

Jumelage et brochage

Le jumelage est basé sur le **protocole Firmata**. Il faut téléverser le programme **StandardFirmata** (IDE Arduino) vers la carte Arduino afin

- qu'elle transmette les ordres de l'ordinateur vers les actionneurs,
- qu'elle remonte les compte-rendus des capteurs vers l'ordinateur.

Dans le script Python le **jumelage est activé** par la commande `jumeau(brochage)` . `'brochage'` est un dictionnaire faisant le lien entre les composants numériques (objet 3D) et les composants réels :

```
brochage={'composant_num':['type',broche,'mode']}.
```

- **a** pour analogique
- **d** pour binaire (digital)

numéro de la broche (0 à 13)

- **i** pour input (entrée)
- **o** pour output (sortie)
- **p** pour pwm (sortie variable)

Par exemple :

```
brochage={'bp_auto':['d',2,'i'],'voy_auto':['d',3,'o']}.
```

Le composant 3D `bp_ext` est associé à la broche **2** en mode **entrée**

Le composant 3D `gyr` est associé à la broche **3** en mode **sortie**

Carte de référence du volet roulant

Pupitre :
- Bouton poussoir monter le volet manuellement : `bp_m()`
- Bouton poussoir arrêter le volet : `bp_a()`
- Bouton poussoir descendre le volet manuellement : `bp_d()`
- Bouton poussoir passer dans le mode automatique : `bp_auto()`
- Voyant mode automatique activé : `voy_auto()`

Capteur de fin de course :
- Volet en position haute : `fdc_h()`
- volet en position basse : `fdc_b()`

Capteur luminosité :
- Dépassement du seuil : `lum()`

Moteur :
- Monter le volet : `mot_m(ordre)`
- Descendre le volet : `mot_d(ordre)`

Valeur retournée par les capteurs et les boutons
- `True` : actif
- `False` : inactif

Brochage (composants numériques) **:**
`'bp_m'`,`'bp_a'`,`'bp_d'`,`'fdc_h'`, `'fdc_d'`,`'mot_o'`,`'mot_f'`, `'bp_auto'`,`'voy_auto'` **et** `'lum'`.

Ordre pour les actionneurs
- `True` : activer
- `False` : désactiver