

	<b>Séquence 3</b> <i>Algorithme et programmation</i>	<b>Activité</b>
	<b>Programmation d'un volet roulant</b>	<b>Sciences de l'Ingénieur</b>
	3A-VOLET- Activité.odt	09/02/2023

## 1. PROLÉMATIQUE

Une entreprise de conception et de fabrication de volet roulant souhaite améliorer ses produits en y intégrant les fonctions communicantes (Internet des Objet - IoT) basées sur une technologie open source de type Arduino.

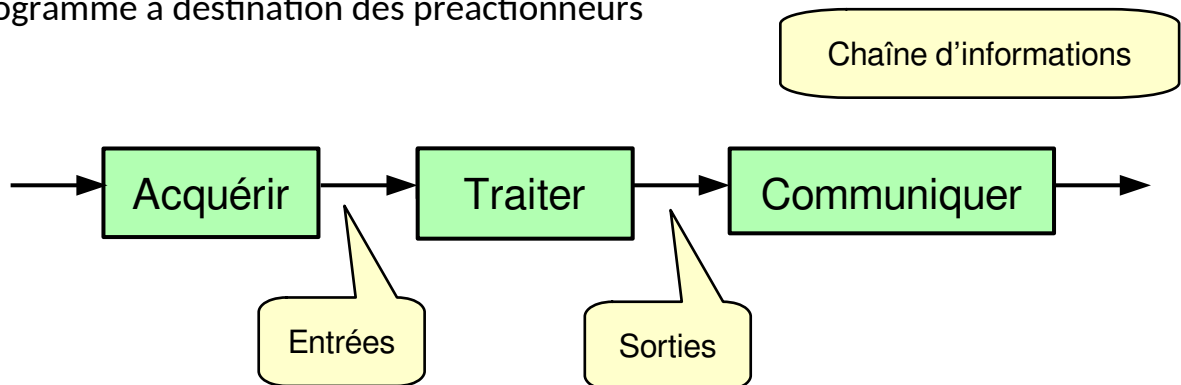
L'objectif est d'élaborer le programme du volet roulant dans le langage Python.



## 2. ENTRÉES - SORTIES

Dans un tout premier temps, il faut faire l'inventaire des entrées et des sorties :

- **Entrées** : informations qui « entrent » dans le programme, informations nécessaires au programme pour définir les actions à faire : les capteurs, les élément du pupitre manipulé par l'utilisateur (boutons, manette, ...)
- **Sorties** : informations qui « sortent » du programme, informations générées par le programme à destination des préactionneurs



A partir de la maquette numérique compléter les croquis avec le nom des entrées-sorties. Puis, compléter le tableau des entrées - sorties :

- nom de la variable,
- description,
- type du signal : binaire, analogique ou numérique.

**Faire valider le tableau par l'enseignant.**

### **3. ANALYSE (DIAGRAMME D'ÉTATS)**

Avant d'établir le programme, vous allez formaliser le fonctionnement du monte-charge à travers son diagramme d'états (SysML). Le monte-charge peut être dans 3 états différents :

- En attente
- Monter le volet
- Descendre le volet

Rappel : Le diagramme d'état du langage SysML vous est rappelé dans le document ressource « **Document de Cours : Diagramme d'états** ».

**Compléter le diagramme d'états en définissant les actions de chaque état.** Les actions correspondent aux sorties.

Rappel : les actions associées à un état sont à définir à l'intérieur du cadre de l'état.

**Compléter le diagramme d'états avec les transitions.** Les transitions sont des équations logiques utilisant les entrées.

Rappel : les transitions sont les flèches qui permettent de passer d'un état à l'autre.

**Faire valider le diagramme d'état par l'enseignant.**

### **4. ALGORIGRAMME**

Grace à l'élaboration du diagramme d'état, nous avons une bonne idée du fonctionnement du volet roulant.

**Établir l'algorithme qui permet la montée et la descente du volet.**

**Faire valider l'algorithme par l'enseignant.**

## 5. CODAGE EN PYTHON

### 5.1 Installation de l'environnement de programmation

Notre environnement de programmation est le langage **Python**, l'éditeur **Spyder** et le jumeau numérique du portail. Afin de l'installer utiliser le **document ressource « Document Technique : Jumeau numérique - Volet roulant »**.

### 5.2 Définition des broches (entrées-sorties)

**A partir de votre tableau des entrées-sorties, proposer un câblage sur le document réponse.** Si le composant Grove utilise le signal primaire et le signal secondaire du port préciser-le en notant les deux broches. **Puis coder la définition des broches dans votre programme.**

Le système Grove vous est présenté dans le **document ressource « Document Technique : Grove pour Arduino »**.

Le codage Python de la définition des broches Arduino se fait par un dictionnaire. Sa définition vous est présenté dans le **document ressource « Document Technique : Jumeau numérique - Volet roulant »**.

### 5.3 Notion de boucle principale

Un microcontrôleur (carte Arduino) n'est jamais à l'arrêt, son fonctionnement repose sur une boucle infinie appelée « **boucle principale** ». Cette boucle permet de placer le microcontrôleur en attente afin d'être à l'écoute des événements susceptibles d'arriver (appui sur un bouton, détection d'un phénomène par un capteur, ...).

**Afin de prendre en main l'environnement de programmation, programmer dans la boucle principale l'allumage du voyant témoin du mode automatique quand le bouton du mode automatique est appuyé.**

**Tester votre programme.**

**Câbler la carte avec les deux composants Grove : un bouton et une led. Attention ! Les composants électroniques sont des objets fragiles. Toute dégradation fera l'objet d'un rachat à la charge de la famille de l'élève.**

**Tester votre programme avec le jumelage sur les deux composants Grove.**

**Puis écrire votre programme sur le document réponse (noter le dictionnaire « jumelage » limité aux deux composants Grove).**

### 5.3 Programme de fonctionnement normal uniquement avec le mode manuel

A partir de votre algorithme (partie 4), écrire le programme du fonctionnement normal du volet roulant. Puis tester votre programme hors ligne (sans jumelage, donc sans les composants Grove).

## 6. MODE AUTOMATIQUE

### 6.1 Sans temporisation

Nous souhaitons maintenant ajouter le mode automatique.

**Modifier votre programme, puis tester le.**

### 6.2 Avec temporisation

Afin d'éviter des montées et des descentes intempestives provoquées par la sensibilité du capteur ou des micro-changements de luminosité, il faut prendre compte le luminosité sur une plus longue durée.

**Il faut donc ajouter à votre programme une temporisation de 2 secondes (durée qui correspond à 5 minutes sur un système réel) sur la prise en compte de la luminosité.**

**Faire valider votre programme par l'enseignant.**

### 6.3 Jumelage

**Câbler la carte avec les éléments Grove. Attention ! Les composants électroniques sont des objets fragiles. Toute dégradation fera l'objet d'un rachat à la charge de la famille de l'élève.**

**Tester votre programme sur le jumeau réel. Imprimer votre programme.**