	Séquence 3 <i>Algorithme et programmation</i>	Activité Sciences de l'Ingénieur
	Programmation d'un portail automatique	
	3A-PORTAIL- Activité.odt	

1. PROBLÉMATIQUE

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

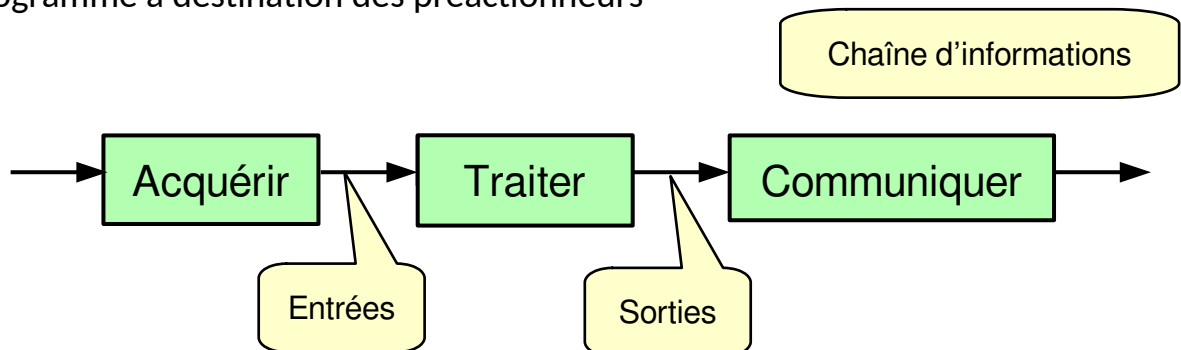
L'objectif est d'élaborer le programme du portail coulissant dans le langage Python.



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 réelle 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,
- le numéro du fil (bague)

Le brochage Arduino sera à compléter plus-tard.

3. ANALYSE (DIAGRAMME D'ÉTATS)

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

- En attente
- Ouvrir le portail
- Fermer le portail

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

3.1 Les valeurs des sorties

Compléter le diagramme d'états en définissant pour chaque état la valeur des sorties.

Par exemple pour une led :

- **led=0** pour éteindre la led,
- **led=1** pour allumer la led.

Rappel : les valeurs des sorties sont à noter à l'intérieur du cadre de l'état.

3.2 Les équations logiques des transitions

Répondre aux questions relatives aux conditions pour passer d'un état à l'autre :

- Quelles sont les conditions pour ouvrir le portail ?
- Lors de l'ouverture, quelle est la condition pour arrêter le mouvement ?
- Quelle est la condition pour fermer le portail ?
- Lors de la fermeture, quelle est la condition pour arrêter le mouvement ?
- Enfin lors de la fermeture, quelle est la condition pour ré-ouvrir le portail ?

Pour regrouper plusieurs conditions dans une même équation, il faut utiliser les opérateurs **OU** et **ET**:

- **OU** : seulement une des deux conditions est nécessaire,
- **ET** : les deux conditions sont nécessaires.

Par exemple, pour une machine à café :

- pour servir un café il faut pour que l'eau soit chaude et qu'on ait appuyé sur le bouton : **eau_chaude = 1 ET bouton=1**,
- si on ajoute le choix petit et grand café : **eau_chaude = 1 ET (bouton_petit=1 OU bouton_grand =1)**.

Définir les équations logiques des 5 transitions. Faire valider votre tableau des entrées-sorties et vos équations des transitions par l'enseignant.

Compléter le diagramme d'états avec les équations logiques des transitions. Rappel : les transitions correspondent aux flèches qui permettent de passer d'un état à l'autre.

4. ALGORIGRAMMES

Grace à l'élaboration du diagramme d'état, nous avons une bonne idée du fonctionnement du portail. Afin d'établir l'algorigramme complet du portail, on se propose de le construire à travers plusieurs étapes.

4.1 Ouvrir (sans arrêt) la barrière avec la télécommande

Établir l'algorigramme qui permet l'ouverture du portail avec le bouton de la télécommande. Ne rien faire si il n'y a pas d'appui. Pour ce programme, il faudra utiliser la structure conditionnelle.

4.2 Ouvrir avec arrêt la barrière avec la télécommande

Établir l'algorigramme qui permet l'ouverture du portail avec le bouton de la télécommande et de l'arrêter en fin de course. Pour ce programme, il faudra utiliser une structure conditionnelle et une boucle indéfinie.

Faire valider les algorigrammes par l'enseignant.

4.3 Ouvrir la barrière puis fermer la barrière (cycle normal sans la sécurité)

Établir l'algorigramme qui permet l'ouverture du portail puis sa fermeture. La gestion de la sécurité avec le capteur barrage n'est pas à traiter.

Faire valider les algorigrammes par l'enseignant.

4.4 Programme complet

En utilisant l'algorigramme précédent, construire l'algorigramme complet.

La gestion de la sécurité doit être prévue. Pendant la fermeture de la barrière : l'appui sur un bouton ou la détection d'un obstacle avec le capteur barrage doit commander la ré-ouverture du portail.

Faire valider votre algorigramme 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 - portail coulissant** ».

5.2 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 gyrophare quand le bouton coté rue est appuyé. Tester votre programme, puis écrire votre programme sur le document réponse.

5.3 Programme de fonctionnement normal sans la sécurité

A partir de votre algorithme (partie 4.3), écrire le programme du fonctionnement normal du portail sans la sécurité. Puis tester votre programme.

5.4 Définition des broches (entrées-sorties) et jumelage

Il faudra définir le brochage de la maquette sur le boîtier Arduino. En fonction du type d'entrées-sorties définir la broche utilisée, **câbler la maquette au boîtier Arduino Uno** puis compléter le tableau des entrées-sorties (partie 2). Les noms des broches de la carte Arduino Uno sont indiqués sur le **document ressource « Document Technique : Broches Arduino Uno »**.

A partir de votre tableau des entrées-sorties coder la définition des broches dans votre programme (porcoul_cmd.py). Le codage Python de la définition des broches Arduino vous est présenté dans le document ressource « Document Technique : Jumeau numérique - portail coulissant ».

Faire valider votre programme par l'enseignant.

Tester votre programme sur le jumeau réel.

5.5 Programme de fonctionnement normal avec la sécurité

A partir de votre algorithme (partie 4.4), écrire le programme du fonctionnement normal du portail avec la sécurité. Puis tester votre programme.