LYCÉE LÉONARD DE VINCI 5A-MONTCHG	**Séquence 5** *Révision - Algorithme et programmation* **Programmation d'un monte-charge**	**Activité** **Sciences de l'Ingénieur**
	5A-MONTCHG- Activité.odt	08/02/2023

1. PROBLÉMATIQUE

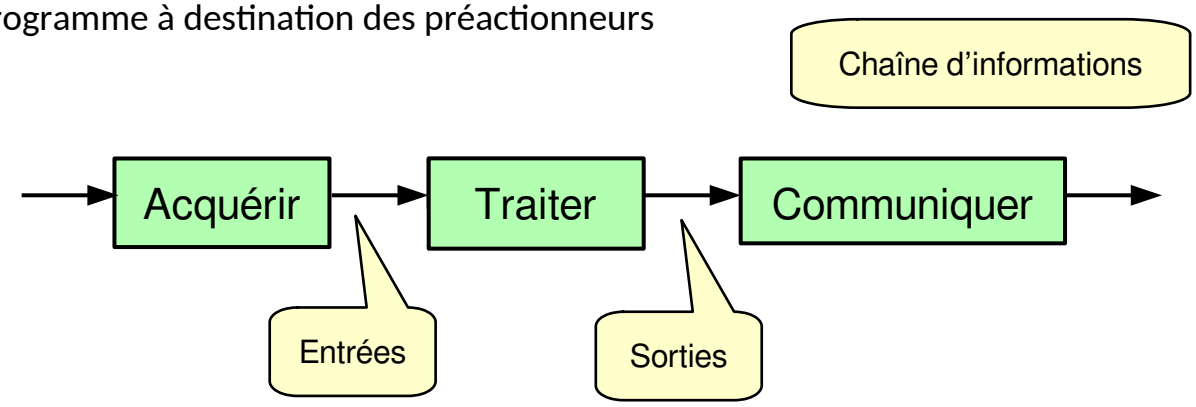
Une entreprise de conception et de fabrication de monte-charge (de 50 à 300 kg) souhaite améliorer ses produits en y intégrant les fonctions communicantes (Internet des Objet – IoT) basées sur une technologie open source de type Arduino.

L'objectif est d'élaborer le programme du monte-charge 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 la cabine
- Descendre la cabine

La mémorisation des appels n'est pas à traiter.

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 monte-charge.

Établir l'algorigramme qui permet la montée et la descente de la cabine. Comme pour le diagramme d'état, la mémorisation des appels n'est pas à traiter.

Faire valider l'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 - Monte-charge** ».

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 évènements 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 d'un voyant quand un bouton est appuyé. Tester votre programme, puis écrire votre programme sur le document réponse.

5.3 Programme de fonctionnement normal

A partir de votre algorigramme (partie 4), écrire le programme du fonctionnement normal du monte-charge. Puis tester votre programme.

Faire valider votre programme par l'enseignant.

6. MÉMORISATION DES APPELS

6.1 Sans la temporisation

Nous souhaitons maintenant prendre en compte les appels quand la cabine est encore en mouvement. Dans un premier temps nous ne prendrons pas en compte la temporisation des 2 secondes lorsque la cabine est arrivée à un niveau.

Proposer un programme.

6.2 Avec la temporisation

Ajouter à votre programme la temporisation.

Faire valider votre programme par l'enseignant.