1STI2D SIN - Exercices Algorithmie

Sur un Arduino UNO, on branche :

- Un capteur, simulé par un bouton poussoir, connecté sur la pin 7 en logique normale (état logique « 1 » quand pressé/lors de la détection)
- Un actionneur, simulé par une LED, connecté à la pin 8 en logique inversé (actif à l'état logique « 0 »)

Comptage

Le capteur détecte la présence de boîtes sur un tapis roulant commandé par le µC. Lorsque 10 boites sont passés devant le capteur, le tapis roulant s'arrête durant 10 secondes, puis redémarre.

Recommandations :

- Démarrer le tapis à l'initialisation de l'arduino
- Utiliser une variable etatCapteur, et stocker l'état de la pin 7 à chaque exécution de la boucle
- Utiliser une variable nbBoites, qui sera incrémentée (nbBoites=nbBoites+1 ou nbBoites++) lors de la détection d'une boite
- Veillez à ce que le code ne s'exécute que lorsque l'entrée ou est reliée le capteur change d'état, en comparant au préalable avec l'état précédent stocké dans etatCapteur
- Utiliser la liaison série pour afficher l'état du compteur (Serial.begin(9600) ; Serial.print("'Etat compteur : ") ; Serial.println(nbBoites)

Front descendant

Le tapis roulant doit s'arrêter durant 2 secondes, puis redémarrer lorsqu'une boîte a été détectée par le capteur, et qu'elle a dépassé ce dernier

Alarme bourrage

le tapis roulant doit s'arrêter si le capteur détecte la même boite durant plus de 6 secondes

Recommandations : la fonction interne de l'Arduino « millis() » permet d'obtenir la durée d'exécution du programme. On peut ainsi « séquencer » le programme, en comparant la valeur renvoyée par la fonction avec une valeur précédemment stockée. Exemple de mise en œuvre (clignotement bref d'une LED toutes les 3 secondes) :

```
unsigned long int lastTime = 0;
void loop() {
    if (millis() - lastTime >= 3000) {
        lastTime = millis();
        digitalWrite(LED,HIGH);
        delay(200);
        digitalWrite(LED,LOW);
    }
}
```


1STI2D SIN - Exercices Algorithmie

Sur un Arduino UNO, on branche :

- Un capteur, simulé par un bouton poussoir, connecté sur la pin 7 en logique normale (état logique « 1 » quand pressé/lors de la détection)
- Un actionneur, simulé par une LED, connecté à la pin 8 en logique inversé (actif à l'état logique « 0 »)

Comptage

Le capteur détecte la présence de boîtes sur un tapis roulant commandé par le µC. Lorsque 10 boites sont passés devant le capteur, le tapis roulant s'arrête durant 10 secondes, puis redémarre.

Recommandations :

- Démarrer le tapis à l'initialisation de l'arduino
- Utiliser une variable etatCapteur, et stocker l'état de la pin 7 à chaque exécution de la boucle
- Utiliser une variable nbBoites, qui sera incrémentée (nbBoites=nbBoites+1 ou nbBoites++) lors de la détection d'une boite
- Veillez à ce que le code ne s'exécute que lorsque l'entrée ou est reliée le capteur change d'état, en comparant au préalable avec l'état précédent stocké dans etatCapteur
- Utiliser la liaison série pour afficher l'état du compteur (Serial.begin(9600) ; Serial.print("'Etat compteur : ") ; Serial.println(nbBoites)

Front descendant

Le tapis roulant doit s'arrêter durant 2 secondes, puis redémarrer lorsqu'une boîte a été détectée par le capteur, et qu'elle a dépassé ce dernier

Alarme bourrage

le tapis roulant doit s'arrêter si le capteur détecte la même boite durant plus de 6 secondes

Recommandations : la fonction interne de l'Arduino « millis() » permet d'obtenir la durée d'exécution du programme. On peut ainsi « séquencer » le programme, en comparant la valeur renvoyée par la fonction avec une valeur précédemment stockée. Exemple de mise en œuvre (clignotement bref d'une LED toutes les 3 secondes) :

```
unsigned long int lastTime = 0;
void loop() {
    if (millis() - lastTime >= 3000) {
        lastTime = millis();
        digitalWrite(LED,HIGH);
        delay(200);
        digitalWrite(LED,LOW);
    }
}
```