

## SÉQUENTIEL

### 1 Panneau solaire asservi

Le rendement d'un panneau solaire peut être sensiblement amélioré en le plaçant sur un support orientable et motorisé, de façon à rester face au soleil. Un capteur fixé sur le panneau, constitué d'un cache et de deux photo-résistances montées en pont diviseur de tension, fournit une tension variable en fonction de l'orientation du panneau face au soleil (voir FIGURE 1). Lorsque les deux résistances sont également éclairées, le capteur renvoie 2,5 V, et lorsqu'une des résistances est à l'ombre du cache, le capteur renvoie une valeur de 1 V ou 4 V selon la résistance à l'ombre.

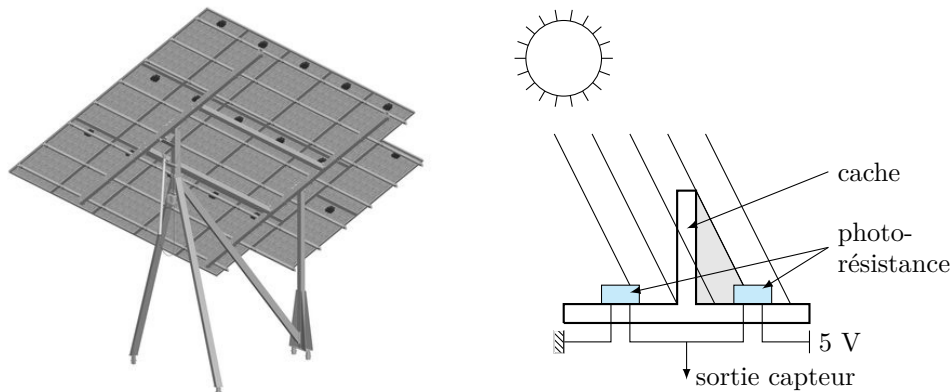


FIGURE 1 – Panneau solaire orientable et représentation du capteur d'orientation.

Le système de commande doit positionner le panneau solaire de façon à maintenir le panneau face au soleil. Un micro-contrôleur permet de piloter le panneau solaire. La commande de mouvement se fait par deux consignes logiques  $M+$  et  $M-$  envoyées en entrée de l'interface de puissance du moteur.

Deux capteurs de fin de course  $C+$  et  $C-$  détectent les positions extrêmes permises pour le mouvement du panneau solaire.

On note  $seuil-$  la tension minimale mesurée sur le capteur lumineux à partir de laquelle le moteur doit tourner ( $M+$ ). On note  $seuil+$  la tension maximale à partir de laquelle le moteur doit tourner dans l'autre sens ( $M-$ ).

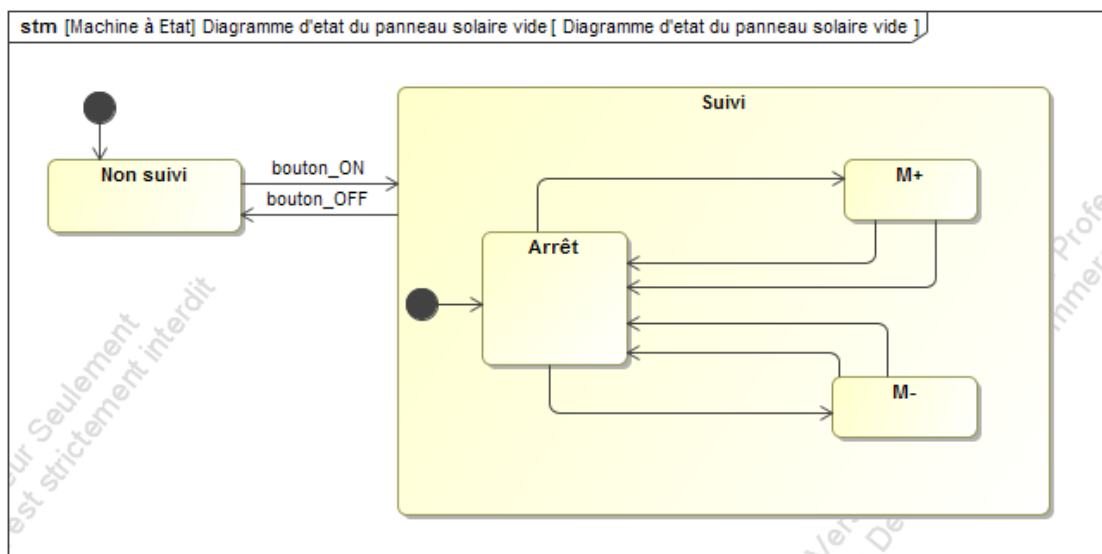
Le cahier des charges est le suivant :

- Un bouton poussoir mono-stable permet, sur le boîtier de commande, d'activer ou désactiver le suivi du soleil par une pression simple.

- En phase de suivi, une LED s'allume sur le boîtier de commande. Elle est éteinte en phase d'arrêt.
- Lorsque la tension capteur *lux* passe sous un seuil limite *seuil-*, le moteur tourne dans le sens *M+* pendant au moins 2 secondes pour éviter les instabilités au voisinage du seuil.
- Lorsque la tension capteur *lux* passe au dessus d'un seuil limite *seuil+*, le moteur tourne dans le sens *M-* pendant au moins 2 secondes.
- Le moteur doit s'arrêter si une fin de course est rencontrée. Il peut néanmoins revenir en arrière.

**Question 1** Déterminer la liste des entrées et sorties du micro-contrôleur ainsi que leur nature (logique ou analogique).

**Question 2** Compléter le diagramme d'états spécifiant le comportement du panneau solaire en renseignant les événements de l'état composite Suivi.



Le programme implanté dans le micro-contrôleur boucle en permanence pour savoir dans quel mode il se trouve et à l'intérieur du mode Suivi ce que le panneau doit faire. En python le programme principal serait le suivant :

```

1 while true :
2     if bouton==1 :
3         led=1
4         suivi()
5     else :
6         led=0
7         non_suivi()

```

On s'intéresse uniquement au mode Suivi. La lecture du niveau lumineux se fait par la commande `lux.read()` qui renvoie la tension mesurée sur le capteur.

**Question 3** Proposer un algorithme en Python correspondant au mode Suivi. On utilisera les noms des variables en les mettant à 1 ou 0 (exemple  $M+ = 1$  correspond au fonctionnement du moteur dans le sens +). La lecture du temps actuel est faite par la commande `time.time()`.

## 2 Gestion de l'avance papier d'une imprimante

On s'intéresse à la gestion de l'avance du papier dans le cas d'une imprimante jet d'encre. Celle-ci possède un moteur associé à un codeur incrémental précis qui met en rotation un rouleau dans un sens ou dans l'autre. Un ensemble d'engrenages débrayables permet de prendre une feuille et la déplacer jusqu'à la tête d'impression ou de la maintenir et de la faire avancer pendant l'impression.

Le fonctionnement normal de l'avance papier est le suivant.

Lorsque l'impression est lancée, le moteur tourne dans un sens ( $M++$ ) pour faire avancer rapidement la feuille jusqu'à un capteur de présence papier (noté *capt*). La tête d'impression est alors mise en position initiale, le moteur est arrêté pendant ce temps. Puis le moteur tourne dans l'autre sens (la feuille avance toujours!) à vitesse lente ( $M-$ ) par à-coups à chaque fois qu'une ligne d'impression est réalisée. Le déplacement fait par la feuille est spécifié par la résolution d'impression choisie au départ par l'utilisateur. On note  $d_a$  le déplacement à faire et  $d$  le déplacement actuel. Lorsque l'impression est terminée, la feuille sort à vitesse rapide ( $M--$ ).

Si aucune feuille n'arrive au capteur de papier pendant 3s, alors un message d'erreur est envoyé pour avertir qu'il n'y a pas de papier.

Si le moteur force trop (blocage de feuille ou bourrage), un message d'erreur est également envoyé. La détection du blocage est faite en mesurant l'intensité dans le moteur qui ne doit pas être supérieure à une valeur maximale donnée  $I_{max}$ .

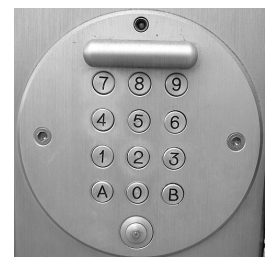
Un diagramme d'état est adapté à la description du fonctionnement de l'imprimante compte-tenu de la spécification indiquée précédemment.

**Question 4** Indiquer les différents états dans lesquels se trouve l'avance papier.

**Question 5** Proposer un diagramme d'états permettant de spécifier le comportement de l'avance papier.

## 3 Digicode

Les digicodes sont très répandus pour contrôler l'accès aux immeubles. Ils permettent de filtrer les entrées sans nécessiter de clés. Il suffit de taper une suite de 4 chiffres ou lettres correspondant au code mémorisé pour que la porte soit déverrouillée.



L'objectif est de réaliser le programme de commande du digicode, dont le cahier des charges est le suivant :

- Le digicode envoie un signal logique à 1 en sortie dès qu'une suite de 4 caractères correspondant au code juste est tapé.
- La porte est déverrouillée pendant 10s, le temps que la personne ouvre la porte, puis se verrouille à nouveau.
- Le code est inscrit dans le programme : la façon de le modifier n'entre pas dans le cadre de cet exercice.

À chaque lecture du digicode, la touche lue est stockée dans un tableau  $T$  de 4 caractères contenant

les 4 dernières touches lues. Ce tableau est comparé au code  $C$  pour décider de l'ouverture ou non de la porte. On utilise un compteur  $N$  initialisé à 0 pour compter les 4 caractères tapés.

Un diagramme d'activité est utile pour spécifier l'ensemble des tâches effectuées par le digicode.

**Question 6** Proposer un tel diagramme en utilisant les variables  $N$ ,  $C$ ,  $T$  ainsi que les actions lecture d'une touche, ouverture, verrouillage.

D'un point de vue fonctionnement, le digicode est constitué d'une matrice de 12 touches et d'un réseau de fils. Sous chacune des touches se situe un interrupteur normalement ouvert, permettant lors d'une pression de connecter deux fils du réseau.

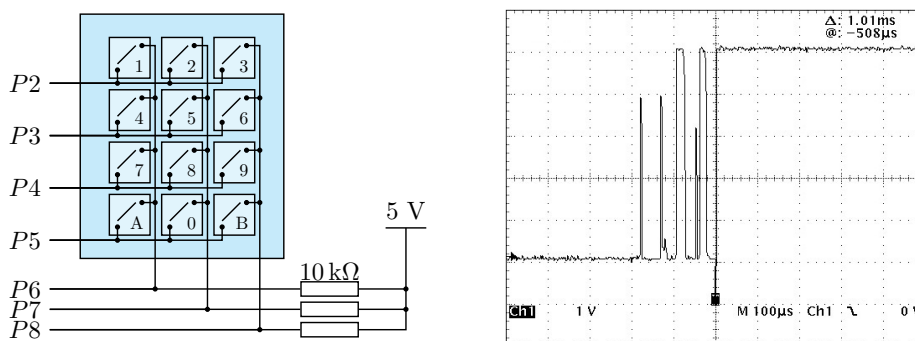


FIGURE 2 – Schéma de principe du digicode ; illustration du phénomène de rebonds.

Les fils sont reliés à 7 entrées logiques du micro-contrôleur. Trois d'entre eux présentent une résistance de tirage (dite pull-up resistance, en anglais) permettant de maintenir par défaut le niveau logique à 1, lorsqu'aucun autre élément n'impose le niveau à 0.

La lecture du clavier se fait ligne par ligne, en analysant les états des colonnes : les lignes ( $P2$  à  $P5$ ) sont toutes mises à 1 sauf la première, mise à 0. Si, lors de la lecture d'une colonne ( $P6$  à  $P8$ ), un niveau logique 0 est détecté, cela signifie qu'une touche de la première ligne est appuyée dans la colonne correspondante. Les lignes suivantes sont lues de la même façon en quelques micro-secondes.

Les interrupteurs présentent un défaut bien connu en électronique : le signal présente durant quelques centaines de micro-secondes des rebonds de basculement de 0 à 1. Pour éviter de lire une information erronée à partir de la détection d'un changement d'état, il est nécessaire d'attendre environ 5 milli-secondes que le signal soit stabilisé avant de lire le clavier.

**Question 7** Répartir les variables  $P2$  à  $P8$  en entrées-sorties pour le micro-contrôleur.

On introduit un tableau représentant l'ensemble des touches réparties par lignes :

```
Liste_touches=[["1","2","3"],["4","5","6"],["7","8","9"],["A","0","B"]]
```

La touche lue est stockée dans une variables `Touche` initialisée à `N`.

Pour lire une variable, on utilise la commande `Pi==1` ou `Pi==0`. Pour imposer une variable à 1 ou 0, on utilise la commande `Pi=1` ou `Pi=0`.

**Question 8** Proposer un programme Python de la fonction lecture en utilisant les variables  $P2$  à  $P8$  et celles introduites ci-dessus.