4. Commande en rotation du panneau photovoltaïque

4.1. Objectif de la séance

Lors de la séance 4, j'ai travaillé avec Hussein sur l'assemblage du code de commande du moteur pas-à-pas et du code de mesure des capteurs de luminosité.
L'objectif était de faire tourner le panneau photovoltaïque en fonction de la différence d'éclairement mesurée par deux résistances photosensibles (LDR), afin d'orienter le panneau vers la direction de lumière maximale.

4.2. Montage des capteurs de lumière

Les deux résistances photosensibles sont montées en série et alimentées par la tension V_{CC}(3,3 V dans notre cas). On réalise ainsi un pont diviseur de tension :

- V_{CC}appliqué à l'entrée du pont ;
- les deux photorésistances en série ;
- prélèvement de la tension au point intermédiaire (entre les deux résistances).

Le principe physique utilisé est le suivant :
plus la photorésistance est éclairée, plus sa résistance diminue. La résistance est donc inversement proportionnelle à la lumière reçue. Cela permet de déduire quel côté du panneau est le plus éclairé à partir de la tension mesurée au milieu du pont diviseur.

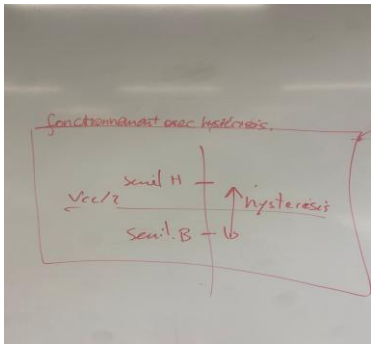
Cette tension est lue par le microcontrôleur (ESP32) grâce à une entrée analogique, ici le GPIO 36.

Dans le code, la tension mesurée est récupérée en millivolts puis convertie en volts :

```
float Vcc = 3.3;

float voltage = analogReadMilliVolts(36);

voltage = 3.3 - voltage / 1000;   // tension sur la première résistance

Serial.println(voltage);

float voltageR2 = 3.3 - voltage;  // tension déduite sur la seconde résistance

Serial.println(voltageR2);
```

À partir de cette mesure, on compare la tension au point milieu à la valeur théorique $V_{CC}/2$pour savoir si une des deux résistances est davantage éclairée que l'autre.

4.3. Principe de commande avec hystérésis



Une première idée, discutée avec Hussein, consistait à exploiter précisément le rapport entre la tension mesurée et $V_{CC}/2$pour calculer un angle de rotation à appliquer au moteur. Cela aurait permis une rotation « continue » en fonction de la différence d'éclairement.
Cependant, l'enseignant nous a conseillé de mettre en place une commande à hystérésis, plus robuste et plus simple à implémenter.

L'idée de l'hystérésis est :

- de définir une tension de référence $V_{ref} = V_{CC}/2$;
- de tolérer une zone « morte » de ±10 % autour de cette valeur (pour éviter les oscillations permanentes du panneau à cause du bruit de mesure ou de petites variations de lumière).

Dans le code, cela se traduit par :

```
float lowerLimit = 0.9 * Vcc / 2;  // seuil bas (90 % de Vcc/2)

float upperLimit = 1.1 * Vcc / 2;  // seuil haut (110 % de Vcc/2);
```

Le comportement souhaité est alors le suivant (pseudo-code fourni en commentaire dans le programme) :

```
/*while(1){

 while(input > Vcc/2 (plus ou moins 10%))

 {

  turn right

  delay 20 mins

  }

while(input < Vcc/2 (plus ou moins 10%))

{
```

turn left

elay 20 mins

}

}

// sundown = system shuts off

*/

Autrement dit :

- si la tension mesurée est significativement supérieure à $V_{CC}/2$, le panneau tourne dans un sens (par exemple vers la droite) ;
- si elle est significativement inférieure à $V_{CC}/2$, le panneau tourne dans l'autre sens (gauche) ;
- tant que la tension reste dans la bande $[lowerLimit, upperLimit]$, le panneau est considéré comme bien orienté et ne bouge pas.

Cette stratégie permet d'éviter les ajustements permanents et d'introduire volontairement une petite inexactitude, acceptable compte tenu de la résolution des mesures.

4.4. Commande du moteur pas-à-pas

en attendant l'intégration complète de la logique d'hystérésis :

float lowerLimit = 0.9 * Vcc / 2;

float upperLimit = 1.1 * Vcc / 2;

À terme, la rotation dépendra de la comparaison entre la tension mesurée

et les seuils lowerLimit / upperLimit.

4.5. Problèmes rencontrés et perspectives

Le principal problème restant concerne le comportement du moteur pas-à-pas : occasionnellement, le moteur ne tourne pas correctement, même lorsque la configuration des broches est correcte. Dans une séance précédente, une erreur de câblage sur les pins avait été

identifiée, mais, malgré la correction, certaines anomalies persistent (blocage, perte de pas, etc.). Il sera donc nécessaire d'investiguer d'autres causes possibles :

- alimentation insuffisante ou instable ;
- couple résistant trop élevé (charge mécanique du panneau) ;
- mauvais choix de séquence de pas ou de vitesse ;
- limitations de la bibliothèque Stepper avec l'ESP32.

Une fois ces problèmes résolus et la logique d'hystérésis effectivement codée et testée, la partie « rotation automatique du panneau » pourra être considérée comme terminée. Nous pourrons alors passer à l'étape suivante : le design du système sur EAGLE , que j'ai déjà commencé à explorer à domicile pour me familiariser avec l'outil.