Suiveur solaire asservi - Rapport individuel

Grégoire MAHON & Romain REVILLOT & Armand LELONG  
Polytech Sorbonne EI2I3-II (groupe B)  
Projet Système Électronique et Informatique

Table des matières

[IV. Principe de fonctionnement des différents composants et étude de leurs caractéristiques 3](#_Toc136888270)

[Question 1 : Système complet (version préliminaire) 3](#_Toc136888271)

[Question 2 : Cellule photovoltaïque et panneau solaire 5](#_Toc136888272)

[A) La cellule photovoltaïque (PV) : 5](#_Toc136888273)

[B) Le panneau solaire : 7](#_Toc136888274)

[Question 3 : Carte STM32 8](#_Toc136888275)

[Question 4 : La tourelle 9](#_Toc136888276)

[Question 5 : Photorésistance 10](#_Toc136888277)

[Question 6 : Schéma complet du montage 11](#_Toc136888278)

[Simulation 11](#_Toc136888279)

[Question 7 : Photorésistance 11](#_Toc136888280)

[1) Cas de R\_LDR constante (intensité lumineuse fixe) 11](#_Toc136888281)

[2) Cas de R\_LDR variable (modélisant une intensité lumineuse variable) 13](#_Toc136888282)

[3) Utilisation d’une source de tension variable pour modéliser une intensité lumineuse variable 13](#_Toc136888283)

[Question 8 : Circuit électrique 14](#_Toc136888284)

[1) Relation du pont diviseur de tension pour trouver la résistance idéale 14](#_Toc136888285)

[2) Montage électronique et test des composants 15](#_Toc136888286)

[VI. Montage de la tourelle 16](#_Toc136888287)

[Question 9 : Tourelle 16](#_Toc136888288)

[3) Test de la tourelle 16](#_Toc136888289)

[4) Manipulation second axe 16](#_Toc136888290)

[5) Rapport cyclique 16](#_Toc136888291)

[VII. Manipulations 17](#_Toc136888292)

[Question 10 : Prise en main de la carte 17](#_Toc136888293)

[Question 11 : Premiers essais avec la carte 17](#_Toc136888294)

[A) Génération de deux signaux PWM 17](#_Toc136888295)

[B) Communication série 19](#_Toc136888296)

[C) Acquisition des données 20](#_Toc136888297)

# IV. Principe de fonctionnement des différents composants et étude de leurs caractéristiques

## Question 1 : Système complet (version préliminaire)

1. Voici un diagramme approximatif du système :

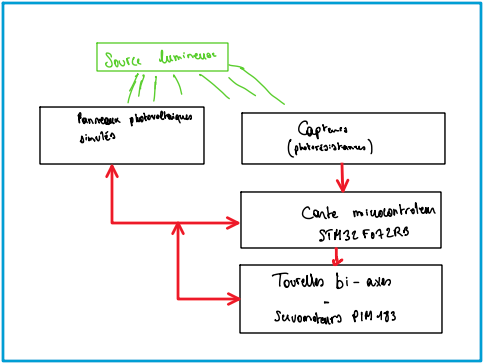


Figure 1 : Diagramme approximatif du système

1. Cahier des charges global du système

**Système de production d'énergie électrique solaire asservi à la source lumineuse en mouvement :**

1. Objectif du système :

Le système est conçu pour produire de l'énergie électrique grâce à un panneau photovoltaïque, en suivant la source lumineuse (simulée).

2. Fonctionnalités requises :

a. Mesure de la luminosité avec les capteurs photorésistances.

b. Contrôle de la tourelle pour maintenir le panneau photovoltaïque dans l'axe normal par rapport à la source lumineuse.

c. Enregistrement des données de mesure et de contrôle pour analyse future.

d. Alimentation autonome grâce au panneau photovoltaïque.

3. Matériel requis :

a. Carte STM32F072RB pour la gestion du système.

b. Tourelle avec deux axes et servomoteurs PIMORONI PIM 183 pour le mouvement du panneau photovoltaïque.

c. Panneau solaire SUNTECH STP005B-12/DEA pour la production d'énergie.

d. Trois capteurs photorésistances NSL-19M51 pour la mesure de la luminosité.

4. Exigences en matière de fiabilité :

a. Le système doit être capable de fonctionner correctement dans des conditions météorologiques variées (pluie, nuages, vent).

b. Le système doit être capable de suivre la source lumineuse avec une précision suffisante pour maximiser la production d'énergie.

c. Le système doit être conçu pour une utilisation durable avec une maintenance minimale.

5. Exigences en matière de sécurité :

a. Le système doit être sécurisé pour éviter de mettre en danger les biens et les personnes.

b. Les dispositifs de protection électrique appropriés doivent être inclus pour éviter les risques d'électrocution ou d'incendie.

6. Autres exigences :

a. Le système doit être économique pour la production d'énergie par rapport aux coûts d'installation et de maintenance.

## Question 2 : Cellule photovoltaïque et panneau solaire

### La cellule photovoltaïque (PV) :

1. Une cellule photovoltaïque (PV) est un dispositif qui convertit l'énergie lumineuse directement en électricité.

Son fonctionnement repose sur le principe de la photovoltaïque, qui consiste à générer du courant électrique à partir de la lumière. Elle est constituée de matériaux semi-conducteurs tels que le silicium, qui ont des propriétés spécifiques de conduction.

Lorsque la lumière pointe éclaire la cellule, elle excite les électrons du matériau semi-conducteur, qui se déplacent alors vers les bords opposés de la cellule. En utilisant des couches différentes de matériaux semi-conducteurs, un potentiel électrique est créé, générant ainsi un courant électrique.

Ce courant peut être collecté et utilisé pour alimenter divers appareils électroniques.

On distingue alors différentes caractéristiques pour la cellule photovoltaïque telles que la tension de sortie, qui dépend de la quantité de cellules en série, ou encore le courant de sortie, qui dépend de la surface totale de la cellule et de l’intensité de la lumière. On peut donc parler de rendement, qui lui dépend de la qualité de la cellule, mais également de la longueur d’onde de la lumière qui éclaire la cellule.

1. Il existe différents types de cellules photovoltaïques :

* Cellules monocristallines : fabriquées à partir de cristaux de silicium purs. Ce sont les plus couramment utilisées car elles sont les plus efficaces en termes de conversion d'énergie solaire en électricité.
* Cellules polycristallines : fabriquées à partir de plusieurs cristaux de silicium fondus ensemble. Elles sont moins efficaces que les cellules monocristallines, mais plus abordables.
* Cellules à base de silicium amorphe : fabriquées à partir de couches minces de silicium amorphe. Elles sont les moins efficaces de toutes les cellules photovoltaïques, mais sont les plus flexibles et peuvent être utilisées pour des applications dans lesquelles la flexibilité est requise.
* Cellules à base de couches minces : fabriquées à partir de couches minces de matériaux tels que le cuivre, l'indium, le gallium et le sélénium. Elles sont plus efficaces que les cellules à base de silicium amorphe, mais sont également plus coûteuses.
* Cellules à concentration : utilisent des miroirs pour concentrer la lumière solaire sur une petite cellule photovoltaïque très efficace. Ce type de cellules est très efficace, mais nécessite un suivi précis de la source de lumière pour fonctionner correctement.

1. Schéma du circuit équivalent d’une cellule photovoltaïque :

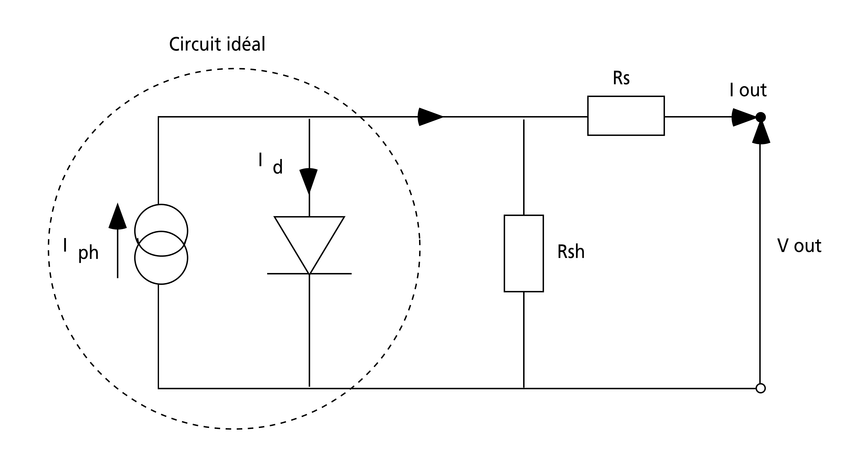


Figure 2 : Circuit équivalent d'une cellule photovoltaïque (source : Djaber Berrian)

1. Paramètres physiques d’une cellule PV :

Le courant généré par la cellule est donné par :

Avec :

* I : courant généré par la cellule photovoltaïque
* I\_L : courant de saturation
* I\_0 : courant de saturation inverse
* q : charge élémentaire
* V est la tension générée par la cellule photovoltaïque
* k : constante de Boltzmann
* T : température absolue

Dans un circuit réel, il faut également tenir compte des résistances internes et d'autres pertes qui peuvent affecter le courant réel généré par la cellule. Cette équation est donc théorique uniquement.

### Le panneau solaire :

1. Le fonctionnement d'un panneau solaire se base sur la conversion de la lumière solaire en énergie électrique grâce à des cellules photovoltaïques (PV).   
   Il existe plusieurs types de cellules photovoltaïques, qui diffèrent par la nature du matériau utilisé pour leur fabrication. Les deux technologies les plus courantes sont :

- Les cellules à base de silicium cristallin (les plus répandues sur le marché)

- Les cellules à base de silicium amorphe (moins courantes mais plus souples)

Le rendement d'un panneau solaire dépend du type de cellules photovoltaïques utilisées et de la qualité de la fabrication. Les cellules à base de silicium cristallin peuvent atteindre des rendements de l'ordre de 15 à 20%, tandis que les cellules à base de silicium amorphe ont un rendement inférieur (entre 5 et 10%).

1. Caractéristiques du panneau solaire :

Modèle : SUNTECH STP005B-12/DEA

- Tension nominale : 12V

- Courant nominal : 0.42A

- Puissance nominale : 5W

- Tension ouverte : 21.6V

- Courant court-circuit : 0.46A

- Tension à la puissance maximale : 17.3V

- Courant à la puissance maximale : 0.29A

- Tension de circuit ouvert : 21.6V

- Courant de court-circuit : 0.46A

- Tension de charge : 18V

- Courant de charge : 0.28A

- Type de cellules : Cellules solaires en silicium monocristallin

- Rendement : 15.5%

- Dimensions : 325 x 250 x 17mm

- Poids : 1.3kg

1. Un tracker solaire est un dispositif qui permet de suivre la progression du soleil au cours de la journée pour maximiser l'exposition de panneaux solaires à la lumière du soleil. Il est utilisé pour augmenter l'efficacité d'un système solaire photovoltaïque en orientant les panneaux solaires vers le soleil à tout moment. Les trackers solaires sont souvent utilisés dans les applications industrielles, les systèmes de production d'énergie décentralisée et les centrales solaires à grande échelle pour augmenter leur rendement énergétique.
2. Il existe deux principaux mécanismes de suivi pour les trackers solaires :

* Le suivi uni axial : Il implique le suivi sur un seul axe, généralement l'axe horizontal pour maximiser la réception de la lumière du soleil.
* Le suivi biaxial : Il implique le suivi sur deux axes, l'un horizontal et l'autre vertical, pour maximiser la réception de la lumière du soleil en fonction de sa position dans le ciel. Ces deux mécanismes peuvent être mis en œuvre en utilisant des moteurs, des servomoteurs ou des actionneurs pour orienter les panneaux solaires en fonction de la position du soleil.

1. Les coordonnées géographiques d'un point sur Terre peuvent être déterminées en utilisant deux mesures : la latitude et la longitude.

La latitude est la mesure de la distance en degrés nord ou sud d'un point par rapport à l'équateur (l'équateur étant considéré comme une ligne de latitude à 0°).

La longitude est la mesure de la distance en degrés est ou ouest d'un point par rapport au méridien de Greenwich (le méridien de Greenwich étant considéré comme une ligne de longitude à 0°).

1. Les angles solaires sont des angles qui décrivent la position relative du Soleil par rapport à un point donné sur Terre. Ils incluent l'angle de latitude, l'angle d'azimut et l'angle d'incidence. L'angle de latitude dépend de la latitude géographique du point d'observation, l'angle d'azimut dépend de la direction du Soleil par rapport à un référentiel cardinal, et l'angle d'incidence dépend de l'angle formé entre la direction du Soleil et la surface perpendiculaire à un panneau solaire.

Les mouvements solaires décrivent la manière dont la position du Soleil évolue dans le ciel au fil de la journée et de l'année.

Les trackers solaires sont donc utilisés pour suivre les mouvements solaires afin d'optimiser l'exposition des panneaux solaires à la lumière du Soleil.

## Question 3 : Carte STM32

1. La carte STM32F072 est une carte microcontrôleur de la famille STM32 produite par STMicroelectronics. Les caractéristiques clés incluent :

* Architecture Cortex-M0+ √† 32 bits
* Fréquence d'horloge maximale de 48 MHz
* Mémoire flash intégrée de 64 KB
* Mémoire RAM intégrée de 8 KB
* Communication : interfaces USART, I2C et SPI
* Entrées/Sorties analogiques (ADC) de 12 bits avec 8 entrées
* Interruptions programmables pour des périphériques externes

Le choix de la carte STM32F072 a été motivé par sa capacité à traiter les entrées/sorties analogiques et les communications avec les périphériques externes. Sa fréquence d'horloge élevée et sa mémoire flash intégrée sont également des critères importants pour les calculs nécessaires au fonctionnement du tracker solaire. En résumé, la carte STM32F072 répond aux contraintes du système en termes d'entrées/sorties analogiques, de capacités de calcul et de communication avec les périphériques externes.

1. Le microcontrôleur STM32F072 peut gérer différents modes énergétiques pour maximiser l'efficacité énergétique du système de tracker solaire. Les modes énergétiques les plus couramment utilisés sont :

* Mode de veille : Ce mode est utilisé pour minimiser la consommation d'énergie lorsque le système est en attente d'une entrée utilisateur ou d'une condition définie. Ce mode permet d'économiser de l'énergie en désactivant les circuits non nécessaires, tels que l'horloge et les périphériques.
* Mode de suspension : Ce mode est similaire au mode de veille, mais la tension est maintenue sur un niveau inférieur pour économiser davantage d'énergie.
* Mode de veille profonde : Ce mode est utilisé pour économiser encore plus d'énergie que les modes de veille et de suspension. La tension est maintenue à un niveau très faible pour minimiser la consommation d'énergie.

L'avantage de ces différents modes énergétiques est qu'ils permettent de maximiser l'efficacité énergétique du système en consommant le moins d'énergie possible lorsque le système n'est pas en cours d'utilisation.

Cependant, le principal inconvénient est qu'il peut y avoir un certain temps de transition entre les modes, ce qui peut entraîner une perte de temps dans la réponse du système. De plus, certains modes peuvent entraîner une perte de données lorsque les circuits sont désactivés.

## Question 4 : La tourelle

1. La tourelle motorisée à 2 axes est un sous-ensemble utilisé pour orienter un système de suivi solaire tel qu'un panneau photovoltaïque en fonction de la position du soleil dans le ciel. Cette tourelle est constituée de deux servomoteurs de référence HS-300 qui permettent des mouvements de rotation selon deux axes, à savoir un axe horizontal et un axe vertical.  
   Les servomoteurs sont contrôlés à l'aide de signaux numériques, généralement envoyés par un microcontrôleur ou un autre dispositif de commande. Le signal de commande est utilisé pour définir la position cible du servomoteur, qui se déplace ensuite vers cette position. Les servomoteurs utilisent des boucles de contrôle en temps réel pour maintenir la position cible, avec une rétroaction qui permet de réguler la vitesse et la direction de rotation.

En ce qui concerne les caractéristiques des servomoteurs HS-300, ils ont un couple de 3,3 kg-cm et une plage de rotation de 180 degrés. Ils sont également alimentés en tension de 4,8V à 6V, et utilisent des signaux de commande de type PWM (modulation de largeur d'impulsion) avec une fréquence de 50 Hz.

1. Pour évaluer l'angle solide couvert par le modèle de tourelle BPT-KT, on prend en compte les angles de rotation possibles pour les deux axes de mouvement, à savoir l'axe horizontal et l'axe vertical.  
   Le servomoteur HS-300 utilisé pour la tourelle a une plage de rotation de 180 degrés, ce qui signifie que chaque axe peut couvrir un angle de 180 degrés. Par conséquent, l'angle solide couvert par la tourelle sera la multiplication de ces deux angles, soit 180 degrés pour l'axe horizontal et 180 degrés pour l'axe vertical, ce qui donne un angle solide total de 32 400 degrés carrés.  
   Cela signifie que la tourelle est capable de couvrir une grande partie du ciel pour suivre la position du soleil et orienter le panneau solaire de manière optimale en fonction de la position du soleil.

## Question 5 : Photorésistance

1. Une photorésistance est un composant électronique qui change de résistance en fonction de l'intensité lumineuse à laquelle elle est exposée.

Le principe de fonctionnement d'une photorésistance est basé sur l'effet photoconductif, qui est la propriété de certains matériaux de devenir plus conducteurs en présence de lumière. Lorsque de la lumière est projetée sur la photorésistance, les électrons dans le matériau sont excités et se déplacent plus facilement, ce qui réduit la résistance de la photorésistance.

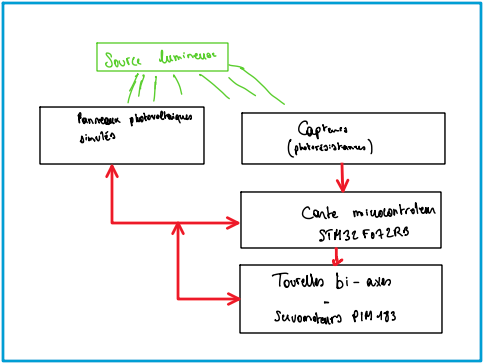
Ainsi, la résistance d'une photorésistance varie en fonction de l'intensité lumineuse à laquelle elle est exposée, ce qui en fait un capteur de lumière utile pour de nombreuses applications, telles que la mesure de la luminosité dans un environnement donné.

1. D'après la documentation du constructeur, les principales caractéristiques des photorésistances NSL-19M51 sont les suivantes :

* Sensibilité spectrale : 540 nm
* Plage de mesure : 0 - 50 000 lux
* Résistance dans l'obscurité : 1 MΩ
* Tension maximale : 150 VDC
* Puissance maximale : 100 mW
* Température de fonctionnement : -30°C à 70°C

1. Pour utiliser une photorésistance avec un amplificateur opérationnel de type suiveur non inverseur, il faut la connecter en série avec une résistance de polarisation. Ce type de montage est adapté pour notre capteur car il amplifie les variations de tension provenant de la photorésistance, et cela sans inverser la polarité.
2. Afin de mesurer une variation spatiale de l’intensité lumineuse, il est préférable d’utiliser plusieurs photorésistances et montages amplificateurs, disposées dans plusieurs positions afin de mesurer les variations d’intensité lumineuses à plusieurs endroits de l’espace.

## Question 6 : Schéma complet du montage



Capteurs avec AOP

Capteurs avec AOP

Capteurs avec AOP

Capteurs

Capteurs

Figure 3 : Schéma du montage

Finalement, le schéma complet reste similaire au diagramme approximatif de la première question.

Cependant, on doit ajouter d’autres capteurs disposés à des endroits différents dans l’espace afin de capter l’intensité lumineuse de façon efficace.

On ajoutera également des amplificateurs avant les photodiodes (non visibles sur le schéma directement, ajoutés dans les blocs capteurs), afin d’amplifier le signal de la photodiode, pour que sa tension puisse être lue convenablement via l’ADC de notre carte nucleo.

# V. Simulation

Par la suite, nous avons effectué des simulations de notre montage sur LTSpice IV afin de valider le dimensionnement des composants et le fonctionnement global du système.

## Question 7 : Photorésistance

### Cas de R\_LDR constante (intensité lumineuse fixe)

On effectue le montage de notre circuit sur LTSpice, avec deux résistances de 5KOhms et un montage amplificateur opérationnel suiveur (pont diviseur de tension).

La simulation LTSpice nous permet alors d’observer les différences de signaux entre l’entrée et la sortie du montage.   
Une image contenant texte, diagramme, ligne, Plan

Description générée automatiquement

Figure 4 : Simulation cas LDR\_R constante (intensité lumineuse fixe)

Une image contenant capture d’écran, logiciel, Logiciel multimédia, Logiciel de graphisme

Description générée automatiquement

Figure 5 : Signaux d'entrée et de sortie simulation photorésistance 1)

On observe sur cette simulation (cf figure 5), que la tension de sortie vaut 2,5V lorsque la tension d’entrée vaut 5V.   
Nous avons donc un gain de 0,5.

### Cas de R\_LDR variable (modélisant une intensité lumineuse variable)

On modifie ici la valeur de R, en la passant en tant que variable dans notre fichier de simulation, cela nous permet ensuite de faire varier sa valeur à notre convenance.   
On fait donc varier sa valeur entre 1 KOhms et 10 KOhms, avec un pas de 1KOhms.

Une image contenant capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Figure 6 : Signaux d'entrée et de sortie avec des valeurs de résistances variables (intensité lumineuse variable)

On remarque dans cette simulation que la valeur de la tension de sortie est proportionnelle à la valeur de R (valeur de nos résistances). On obtient alors une tension de sortie de 3,3V pour une valeur R = 10 KOhms, et une tension de sortie de 0,8V pour une valeur de R = 1 KOhms.

### Utilisation d’une source de tension variable pour modéliser une intensité lumineuse variable

On utilise ici la fonction PULSE disponible sur LTSpice afin de créer une source de tension variable.   
Une image contenant texte, diagramme, ligne, Police

Description générée automatiquement

Figure 7 : Simulation avec source de tension variable

Une image contenant capture d’écran, Logiciel multimédia, Logiciel de graphisme, ligne

Description générée automatiquement

Figure 8 : Simulation avec source de tension variable (PULSE)

On obtient ici le résultat donné en gabarit dans le sujet. Le montage amplificateur suiveur permet ici de ne pas modifier le signal d’entrée, mais uniquement de stabiliser la tension de sortie.

## Question 8 : Circuit électrique

### Relation du pont diviseur de tension pour trouver la résistance idéale

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

Figure 9 : Schéma électrique simplifié (sujet)

On utilise la relation du pont diviseur de tension afin de trouver la valeur de résistance idéale.   
  
Dans ce schéma, la tension V1 vaut 𝑉1 = 𝑉𝑐𝑐 ∗ (𝑅1 / (𝑅1 + 𝑅2) ).

On peut donc en déduire la valeur de R1 qui vaut 5 KOhms.

### Montage électronique et test des composants

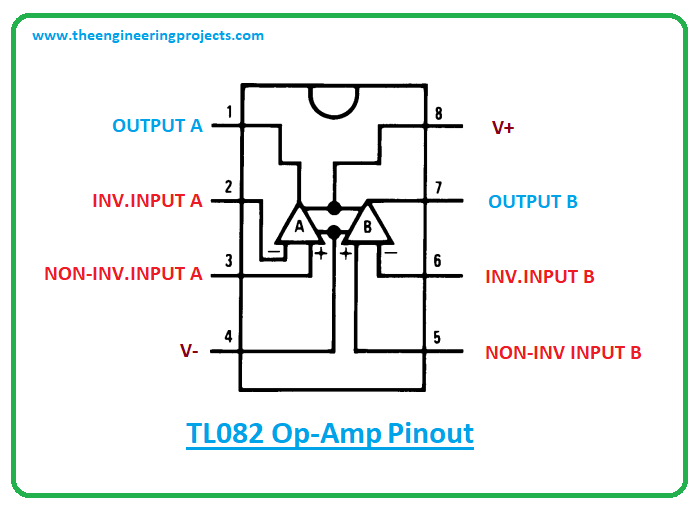
On utilise pour notre montage deux amplificateurs opérationnels TL082, dont le schéma interne est le suivant : 

Figure 10 : Schéma AOP TL082

Nous avons décidé d’utiliser ce composant pour sa simplicité d’utilisation, et car il comporte deux AOP, ce qui nous permet donc d’amplifier les signaux de deux photorésistances avec un seul composant. Nous utilisons alors le second AOP afin de traiter le signal de la troisième photorésistance.

Nous avons ensuite choisi des résistances de 4,7 KOhms afin de réaliser notre pont diviseur de tension (cf partie 1).

Nous avons choisi d'alimenter les AOP avec une tension de +5V / -5V au lieu de 0V / 5V, en utilisant l’Analog Discovery, en mode Power Supply. Cette décision est principalement motivée par le désir d'éviter le phénomène de latch-up.

Le latch-up est un phénomène indésirable qui peut se produire dans les circuits intégrés, notamment les AOP, lorsque la tension d'entrée dépasse la tension d'alimentation. Dans notre cas, si nous alimentions les AOP avec une tension de 0V / 5V, toute tension d'entrée négative pourrait potentiellement déclencher un latch-up. Cela pourrait entraîner une consommation de courant excessive et potentiellement endommager les AOP.

En alimentant les AOP avec une tension de +5V / -5V, nous nous assurons que la tension d'entrée reste toujours dans la plage de tension d'alimentation, même si la tension d'entrée devient négative. Cela nous permet d'éviter le latch-up et de garantir un fonctionnement sûr et fiable des AOP.

De plus, l'utilisation d'une alimentation symétrique de +5V / -5V nous permet de bénéficier d'une gamme complète de tensions de sortie, ce qui est particulièrement utile pour les applications où nous avons besoin de tensions de sortie négatives.

On peut ensuite vérifier la conformité du signal de sortie avec le mode Scope de l’analog discovery, ce qui nous a donné satisfaction.

# VI. Montage de la tourelle

Avant d’utiliser une tourelle, il faut l’assembler. Dans notre cas, elle nous a été fournie livrée.

## Question 9 : Tourelle

Selon la documentation du constructeur, le couple de la tourelle HS-300 est de 3,02 Kg\*cm à

4,8V.

La longueur de l’axe mesure 3,2 cm, Nous disposons d’un panneau solaire pesant 700 grammes.

Le poids max supportable est calculable de la façon suivante :

Pmax = Couple / Longueur

On obtient donc un poids maximal de 0,94Kg.

La tourelle nous ayant été fournie livrée, nous n’avons pas eu d’étape assemblage à réaliser.

## Test de la tourelle

Afin de tester notre tourelle convenablement, nous avons codé un programme simple qui envoie un signal PWM sur le pin en question du servomoteur. Nous avons alors réussi à faire tourner sur les deux axes.

Concernant l’alimentation, nous avons utilisé le +5V avec le powersupply de l’analog discovery.

### Manipulation second axe

La manipulation second axe est exactement la même que sur le premier axe.

### Rapport cyclique

En modifiant le rapport cyclique de notre signal PWM, on peut modifier l’angle de notre servomoteur.

# VII. Manipulations

Nous avons entièrement réalisé le code embarqué en langage C (compilation avec le compilateur embarqué de l’IDE Keil uVision 4).

## Question 10 : Prise en main de la carte

Nous avons dans un premier temps codé un simple programme permettant de faire clignoter une LED sur la carte. Nous avons alors utilisé la bibliothèque, créée lors de nos cours de microcontrôleurs avec M. Darga, que nous avons nommé BSP.c. Cette librairie de fonctions permet d’effectuer diverses opérations telles que l’initialisation des LEDs, l’initialisation de la console etc…  
  
Nous avons alors, pour tester notre carte, utilisé une fonction créée précédemment nommée BSP\_LED\_Toggle() ; qui permet d’inverser l’état de la LED.  
Nous avons ensuite ajouté un délai de 500ms afin de créer un clignotement, après avoir placé ces deux lignes de code C dans une boucle while infinie.

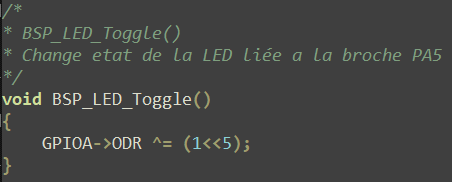


Figure 11 : Détails de la fonction BSP\_LED\_Toggle()

## Question 11 : Premiers essais avec la carte

### Génération de deux signaux PWM

Les trois caractéristiques d’un signal PWM sont la fréquence, la période, et la largeur des impulsions (🡪rapport cyclique), ainsi que l’amplitude du signal (tension).

#### Résolution du signal PWM

Nous avons pris une résolution de 8 bits pour avoir un maximum de précision sur le contrôle des servomoteurs (180 degrés d’amplitude 🡪 8 bits = 256 échantillons).

#### Initier le signal PWM

Afin d’initialiser le signal PWM, nous utilisons les pins 8 et 9 du GPIOA.

Il faut également activer la clock et le timer.   
Voici notre fonction d’initialisation de la PWM :

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 12 : Fonction d'initialisation de la PWM

Nous avons alors un signal PWM généré sur les pins PA8 et PA9 (registre GPIOA).

Il suffit ensuite d’activer ces pins dans notre code principal en utilisant les registres TIM1->CCR1 pour PA9 et TIM1->CCR2 pour PA8.

Il suffit ensuite d’incrémenter ou de décrémenter leur valeur à chaque tour de boucle afin de faire varier l’angle du servomoteur.

### Communication série

Pour ce qui est de la communication UART, nous utilisons également une fonction de notre librairie BSP.c afin d’initialiser la communication série.

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 13 : Initialisation de la communication UART

Nous avons alors utilisé nos fonctions créées lors des cours afin d’afficher d’envoyer des messages via le port série.   
  
On peut ensuite ouvrir un terminal série sur notre machine permettant d’afficher les messages reçus sur le port correspondant au driver USB de la carte, au baudrate de 115200 baud/s (défini dans la fonction figure 13).

### Acquisition des données

#### Type des données / unité

Les données envoyées à la carte sont les valeurs de tensions analogiques des photorésistances (en sortie des AOP).

Cela correspond donc à un signal analogique.  
Afin de lire cette valeur dans notre code, nous ne pouvons uniquement lire des valeurs numériques. Pour cela, nous utilisons donc l’ADC (analog to digital converter). Cela nous permet de convertir la donnée analogique en donnée numérique, afin de lire la valeur de la photorésistance directement avec notre carte.  
Nous pourrons par la suite afficher ces valeurs via le terminal série, comme décrit dans la partie B auparavant.

#### Broches

On utilisera les pins PC0 (ADC\_IN10), PC1 (ADC\_IN11), PC2 (ADC\_IN12).

#### Utilisation de l’ADC (mémoire ?)

Il est possible d’utiliser l’ADC de la carte nucleo sans utiliser de mémoire. On peut simplement lire la valeur et l’afficher directement en lisant simplement la valeur dans le registre GPIOC). Cependant, si nous souhaitons enregistrer la valeur pour la comparer avec une autre, par exemple, dans ce cas il sera nécessaire d’enregistrer cette dernière dans une variable en mémoire afin de pouvoir traiter sa valeur.

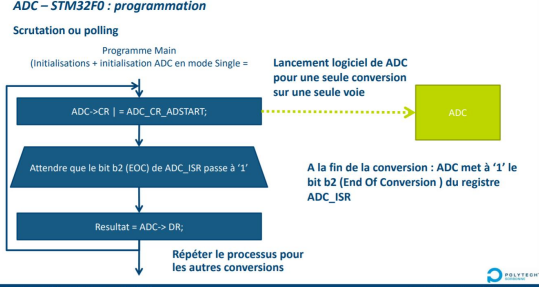
Cependant, si nous n’utilisons pas d’enregistrement en mémoire, il faut attendre que le 2e bit (EOC = End Of Conversion) du registre ADC\_ISR soit à 1 avant de lire la valeur. Voici notre transparent de cours traitant de ce sujet :   


Figure 14 : Transparent de cours sur l'attente de conversion par l'ADC

#### Initialisation de l’ADC

Dans le cadre de notre projet, l'initialisation de l'ADC est une étape cruciale pour la lecture des valeurs de trois photorésistances. Cette initialisation est effectuée dans la fonction principale de notre programme : Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 15 : Initialisation de l'ADC dans le main.c

La fonction **BSP\_ADC\_Init()** est définie dans le fichier "bsp.c", et elle est responsable de la configuration et de l'initialisation de l'ADC sur notre microcontrôleur STM32. Cette fonction configure les paramètres de l'ADC, tels que la résolution, le mode de fonctionnement, la vitesse de conversion, et plus encore, selon les spécifications de notre application.

Une fois que l'ADC est initialisé, nous pouvons commencer à lire les valeurs des photorésistances en utilisant la fonction **Read\_PhotoResistor()**. Cette fonction sélectionne le canal ADC approprié, démarre la conversion, attend que la conversion soit terminée, puis lit et renvoie la valeur convertie.

Nous avons mis en place un système d'acquisition de données pour lire les valeurs de trois photorésistances. Pour ce faire, nous utilisons deux amplificateurs opérationnels (AOP) TL082 et l'ADC du microcontrôleur STM32.

La fonction Read\_PhotoResistor dans notre code est responsable de l'initialisation et de la lecture de l'ADC. Voici comment elle fonctionne :

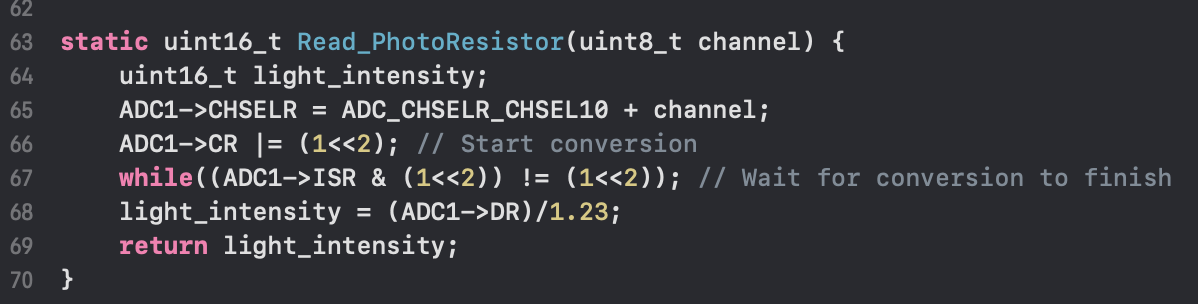


Figure 16 : Fonction de lecture des valeurs des photorésistances

Dans cette fonction, nous commençons par sélectionner le canal ADC approprié en fonction de la photorésistance que nous voulons lire. Ensuite, nous démarrons la conversion ADC en mettant le bit 2 du registre de contrôle ADC (ADC1->CR) à 1. Nous attendons ensuite que la conversion soit terminée en vérifiant le bit 2 du registre de statut ADC (ADC1->ISR). Une fois la conversion terminée, nous lisons la valeur convertie à partir du registre de données ADC (ADC1->DR) et la divisons par 1.23 pour obtenir l'intensité lumineuse. Cette valeur est ensuite renvoyée par la fonction.

## Question 12 : Algorithme

### Fonctions principales à réaliser dans notre algorithme

1. **Initialisation du système** : Le programme doit initialiser le système, y compris la configuration de l'horloge système, l'initialisation de la console, de la LED, du Timer PWM et de l'ADC.
2. **Lecture des photorésistances** : Le programme doit lire les valeurs des trois photorésistances via l'ADC. Ces valeurs sont utilisées pour déterminer l'intensité lumineuse sur chaque photorésistance.
3. **Calcul de l'intensité lumineuse moyenne** : Le programme calcule l'intensité lumineuse moyenne à partir des valeurs lues des deux premières photorésistances.
4. **Ajustement des angles des servomoteurs** : En fonction des valeurs lues des photorésistances et de l'intensité lumineuse moyenne, le programme ajuste les angles des servomoteurs pour orienter correctement le panneau solaire.

### Réalisation de l’algorithme

L'algorithme commence par initialiser le système et les variables d'état. Ensuite, il entre dans une boucle infinie où il lit les valeurs des photorésistances et ajuste les angles des servomoteurs.

Pour chaque photorésistance, l'algorithme sélectionne le canal ADC approprié, démarre une conversion ADC, attend que la conversion soit terminée, puis lit la valeur convertie.

L'algorithme calcule ensuite l'intensité lumineuse moyenne à partir des valeurs des deux premières photorésistances. Il utilise cette moyenne et la valeur de la troisième photorésistance pour ajuster les angles des servomoteurs.

Pour ajuster l'angle du servomoteur horizontal, l'algorithme compare les valeurs des deux premières photorésistances. Si la différence entre elles est supérieure à une certaine tolérance, il augmente ou diminue l'angle du servomoteur en conséquence.

Pour ajuster l'angle du servomoteur vertical, l'algorithme compare l'intensité lumineuse moyenne avec la valeur de la troisième photorésistance. Si la différence entre elles est supérieure à une certaine tolérance, il augmente ou diminue l'angle du servomoteur en conséquence.

Enfin, l'algorithme attend un court délai avant de recommencer la boucle.

Voici **l’algorithme en pseudo-code** :

**Fonction principale :**

🡪 Initialiser le système (ADC, UART, etc…)

🡪 Initialiser les variables d'état

Tant que vrai :

🡪 Lire la valeur de la photorésistance 1

🡪 Lire la valeur de la photorésistance 2

🡪 Lire la valeur de la photorésistance 3

🡪 Ajuster les angles des servomoteurs

🡪 Attendre un court délai

**Fonction Initialiser le système :**

🡪 Configurer l'horloge système

🡪 Initialiser la console

🡪 Initialiser la LED

🡪 Initialiser le Timer PWM

🡪 Initialiser l'ADC

**Fonction Initialiser les variables d'état :**

🡪 Définir l'angle du servomoteur 1

🡪 Définir l'angle du servomoteur 2

🡪 Définir la tolérance horizontale

🡪 Définir la tolérance verticale

**Fonction Lire la valeur de la photorésistance (canal) :**

🡪 Sélectionner le canal ADC approprié

🡪 Démarrer une conversion ADC

🡪 Attendre que la conversion soit terminée

🡪 Lire la valeur convertie

🡪 Retourner la valeur convertie

**Fonction Ajuster les angles des servomoteurs :**

🡪 Calculer l'intensité lumineuse moyenne à partir des valeurs des photorésistances 1 et 2

Si la différence entre les valeurs des photorésistances 1 et 2 est supérieure à la tolérance horizontale :

Si la valeur de la photorésistance 1 est inférieure à celle de la photorésistance 2 et que l'angle du servomoteur 2 est inférieur à 2400 :

🡪 Augmenter l'angle du servomoteur 2

Sinon si la valeur de la photorésistance 2 est inférieure à celle de la photorésistance 1 et que l'angle du servomoteur 2 est supérieur à 600 :

🡪 Diminuer l'angle du servomoteur 2

Si la différence entre l'intensité lumineuse moyenne et la valeur de la photorésistance 3 est supérieure à la tolérance verticale :

Si l'intensité lumineuse moyenne est inférieure à la valeur de la photorésistance 3 et que l'angle du servomoteur 1 est inférieur à 3000 :

🡪 Augmenter l'angle du servomoteur 1

Sinon si l'intensité lumineuse moyenne est supérieure à la valeur de la photorésistance 3 et que l'angle du servomoteur 1 est supérieur à 1000 :

🡪 Diminuer l'angle du servomoteur 1

## Question 13 : Codage logiciel

Voici notre code main.c :

|  |
| --- |
| # @author : Grégoire MAHON |
| # EI2I 3 - II |
|  |
| #include "stm32f0xx.h" |
| #include "bsp.h" |
| #include "delay.h" |
| #include "main.h" |
| #include <math.h> |
|  |
| #define FALSE 0 |
| #define TRUE 1 |
|  |
| // Function Prototypes |
| **static** **void** **SystemClock\_Config**(**void**); |
| **static** **void** **Initialize\_Variables**(**void**); |
| **static** **uint16\_t** **Read\_PhotoResistor**(**uint8\_t** channel); |
| **static** **void** **Adjust\_ServoAngles**(**void**); |
|  |
| // Global Variables |
| **uint16\_t** average\_light\_intensity; |
| **uint16\_t** diff\_light\_intensity\_1\_2; |
| **uint16\_t** diff\_light\_intensity\_12\_3; |
| **uint16\_t** light\_intensity\_1; |
| **uint16\_t** light\_intensity\_2; |
| **uint16\_t** light\_intensity\_3; |
| **uint16\_t** tolerance\_horizontal; |
| **uint16\_t** tolerance\_vertical; |
| **uint8\_t** horizon; |
| **uint8\_t** vertical; |
| **uint16\_t** servo\_angle\_1; |
| **uint16\_t** servo\_angle\_2; |
|  |
| **int** **main**() { |
| // Configure System Clock |
| BSP\_Console\_Init(); |
| mon\_printf("Intialisation de la console OK!**\r\n**"); |
|  |
| // Initialize LED pin |
| BSP\_LED\_Init(); |
| BSP\_TIMER\_PWM\_Init(); |
| BSP\_ADC\_Init(); |
| mon\_printf("Initialisation de l'ADC OK!**\r\n**"); |
|  |
| // Initialize state variables |
| Initialize\_Variables(); |
|  |

|  |
| --- |
| **while**(**1**){ |
| light\_intensity\_1 = Read\_PhotoResistor(**10**); |
| light\_intensity\_2 = Read\_PhotoResistor(**11**); |
| light\_intensity\_3 = Read\_PhotoResistor(**12**); |
| Adjust\_ServoAngles(); |
| BSP\_DELAY\_ms(**1**); |
| } |
| } |
|  |
| **static** **void** **Initialize\_Variables**(**void**) { |
| servo\_angle\_1 = **2100**; |
| servo\_angle\_2 = **1200**; |
| tolerance\_horizontal = **100**; // 100 both |
| tolerance\_vertical = **100**; |
| } |
|  |
| **static** **uint16\_t** **Read\_PhotoResistor**(**uint8\_t** channel) { |
| **uint16\_t** light\_intensity; |
| ADC1->CHSELR = ADC\_CHSELR\_CHSEL10 + channel; |
| ADC1->CR |= (**1**<<**2**); // Start conversion |
| **while**((ADC1->ISR & (**1**<<**2**)) != (**1**<<**2**)); // Wait for conversion to finish |
| light\_intensity = (ADC1->DR)/**1.23**; |
| **return** light\_intensity; |
| } |
|  |
| **static** **void** **Adjust\_ServoAngles**(**void**) { |
| TIM1->CCR1 = servo\_angle\_1; |
| TIM1->CCR2 = servo\_angle\_2; |
|  |
| // Compute average light intensity |
| average\_light\_intensity = (light\_intensity\_1 + light\_intensity\_2) / **2**; |
|  |
| // Adjust horizontal servo angle |
| **if** (light\_intensity\_1 < light\_intensity\_2){ |
| diff\_light\_intensity\_1\_2 = abs(light\_intensity\_2 - light\_intensity\_1); |
| **if**(diff\_light\_intensity\_1\_2 > tolerance\_horizontal){ |
| **if** (servo\_angle\_2 < **2400**){ |
| servo\_angle\_2++; |
| } |
| } |
| } |
| **if** (light\_intensity\_2 < light\_intensity\_1){ |
| diff\_light\_intensity\_1\_2 = abs(light\_intensity\_1 - light\_intensity\_2); |
| **if**(diff\_light\_intensity\_1\_2 > tolerance\_horizontal){ |
| **if** (servo\_angle\_2 > **600**){ |
| servo\_angle\_2--; |
| } |
| } |
| } |
|  |

|  |
| --- |
| // Adjust vertical servo angle |
| **if**(average\_light\_intensity < light\_intensity\_3){ |
| diff\_light\_intensity\_12\_3 = abs(average\_light\_intensity - light\_intensity\_3); |
| **if**(diff\_light\_intensity\_12\_3 > tolerance\_vertical){ |
| **if**(servo\_angle\_1 < **3000**){ |
| servo\_angle\_1++; |
| } |
| } |
| } |
|  |
| **if**(average\_light\_intensity > light\_intensity\_3){ |
| diff\_light\_intensity\_12\_3 = abs(light\_intensity\_3 - average\_light\_intensity); |
| **if**(diff\_light\_intensity\_12\_3 > tolerance\_vertical){ |
| **if**(servo\_angle\_1 > **1000**){ |
| servo\_angle\_1--; |
| } |
| } |
| } |
| } |

Une fois le code source intégré dans notre carte microcontrôleur, nous avons procédé à une série de tests pour vérifier son bon fonctionnement. Ces tests ont été conçus pour couvrir tous les aspects de la fonctionnalité du programme, y compris la lecture correcte des valeurs des photorésistances, le calcul précis de l'intensité lumineuse moyenne, et l'ajustement approprié des angles des servomoteurs.

Pour le test des photorésistances, nous avons exposé chaque photorésistance à différentes intensités lumineuses et vérifié que les valeurs lues par l'ADC correspondaient à nos attentes. Nous avons également vérifié que l'intensité lumineuse moyenne était correctement calculée à partir des valeurs des deux premières photorésistances.  
Pour le test des servomoteurs, nous avons vérifié que les angles des servomoteurs étaient correctement ajustés en fonction des valeurs lues des photorésistances et de l'intensité lumineuse moyenne. Nous avons observé le mouvement des servomoteurs pour vérifier qu'ils se déplaçaient dans la bonne direction et d'un montant approprié en réponse aux changements d'intensité lumineuse.

Tous les tests ont été passés avec succès, ce qui confirme que le programme fonctionne comme prévu.

Nous avons également effectué des tests sur le montage physique du système en plaçant les trois photorésistances sur un morceau de carton, disposées en forme de triangle, au niveau de la partie supérieure de la tourelle. Nous avons ensuite observé le comportement des servomoteurs en exposant les photorésistances à différentes sources lumineuses.

Lors de ces tests, nous avons constaté que les servomoteurs suivaient effectivement la lumière détectée par les photorésistances. Lorsque nous avons déplacé une source lumineuse vers l'une des photorésistances, les servomoteurs se sont ajustés pour orienter la tourelle dans la direction de la source lumineuse. De plus, nous avons remarqué que plus l'intensité lumineuse était élevée, plus les servomoteurs se déplaçaient pour s'aligner avec la source lumineuse.

Ces observations confirment le bon fonctionnement du système dans la détection et le suivi de la lumière à l'aide des photorésistances et des servomoteurs. Le programme et le matériel ont été efficaces pour permettre à la tourelle de s'orienter vers la source lumineuse détectée.

## Question 14 : Régulateur PID

### Principe d’un régulateur PID

Un correcteur PID (proportionnel-intégral-dérivé) est un dispositif utilisé en régulation automatique pour ajuster et contrôler un système en fonction de l'erreur entre la valeur désirée et la valeur réelle (mesurée).

Le correcteur PID utilise trois termes pour calculer la sortie du système de contrôle :

1. Le terme proportionnel (P) est proportionnel à l'erreur actuelle entre la valeur désirée et la valeur mesurée. Il amplifie cette erreur pour générer une correction proportionnelle. Plus l'erreur est grande, plus la correction proportionnelle sera importante.
2. Le terme intégral (I) intègre l'erreur cumulée sur une période de temps pour générer une correction intégrale. Il prend en compte l'historique des erreurs et ajoute une correction en fonction de la somme de ces erreurs accumulées. Cela permet de réduire les erreurs persistantes à long terme.
3. Le terme dérivé (D) est dérivé de la variation de l'erreur par rapport au temps. Il génère une correction basée sur la vitesse à laquelle l'erreur change. Le terme dérivé permet de réagir rapidement aux variations rapides de l'erreur et de prévenir les dépassements et les oscillations excessives.

Le correcteur PID combine ces trois termes pour générer une sortie de commande qui ajuste le système de manière optimale. Le terme proportionnel réagit immédiatement à l'erreur actuelle, le terme intégral élimine les erreurs persistantes et le terme dérivé anticipe les variations de l'erreur.

Les coefficients P, I et D du correcteur PID sont ajustés expérimentalement pour chaque système en fonction de ses caractéristiques et de ses exigences de réponse. Une fois les coefficients optimisés, le correcteur PID peut fournir un contrôle précis et stable du système en minimisant l'erreur entre la valeur désirée et la valeur réelle.

## Implémentation du régulateur PID

Dans notre système de suivi de la lumière, le correcteur PID peut être appliqué pour améliorer la précision et la stabilité des mouvements des servomoteurs en fonction des variations d'intensité lumineuse détectées par les photorésistances.

Voici comment le correcteur PID pourrait être utilisé dans notre système :

1. Mesure de l'erreur : L'erreur serait calculée en comparant la valeur désirée (par exemple, l'intensité lumineuse moyenne souhaitée) avec la valeur réelle (la valeur mesurée à partir des photorésistances). Cette différence représenterait l'erreur qui sera utilisée dans le contrôle PID.
2. Proportionnel (P) : Le terme proportionnel serait utilisé pour ajuster l'angle des servomoteurs proportionnellement à l'erreur. Plus l'erreur est grande, plus l'angle des servomoteurs sera ajusté de manière significative pour réduire l'erreur.
3. Intégral (I) : Le terme intégral prendrait en compte l'accumulation des erreurs au fil du temps. Il permettrait de corriger les erreurs persistantes à long terme en ajustant progressivement l'angle des servomoteurs en fonction de l'erreur cumulée.
4. Dérivé (D) : Le terme dérivé réagirait à la vitesse de variation de l'erreur. Il permettrait de détecter les changements rapides de l'intensité lumineuse et d'ajuster les servomoteurs de manière préventive pour éviter les oscillations excessives ou les dépassements.
5. Réglage des coefficients PID : Les coefficients P, I et D du correcteur PID seraient ajustés expérimentalement pour obtenir les performances souhaitées. Ces coefficients seraient optimisés pour chaque système en prenant en compte les caractéristiques des servomoteurs, des photorésistances et des sources lumineuses.

En utilisant le correcteur PID, notre système de suivi de la lumière pourrait réagir de manière plus précise et rapide aux variations d'intensité lumineuse, en ajustant les angles des servomoteurs de manière optimale. Le suivi de la lumière serait alors plus précis.

## Code avec le PID implémenté

Nous avons dans un premier temps implémenté les coefficients PID en tant que constantes :

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Figure 17 : Constantes coefficients PID

Ensuite, nous avons ajouté les variables pour le contrôle PID (toutes initialisées à 0.0):

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Figure 18 : Variables pour le contrôle PID

Enfin, nous avons implémenté la correction PID dans notre fonction de contrôle des servomoteurs de la tourelle :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Figure 19 : Fonction de contrôle des servomoteurs avec régulateur PID

L'implémentation du régulateur PID dans notre code est réalisée selon les principes classiques du contrôle PID. Les coefficients du régulateur (KP, KI et KD) sont définis en tant que constantes dans notre programme. Nous avons utilisé les valeurs KP = 0.5, KI = 0.2 et KD = 0.1, qui ont été choisies initialement et peuvent être ajustées en fonction des besoins spécifiques de notre système.

Dans la fonction Adjust\_ServoAngles(), nous avons ajouté les étapes supplémentaires pour calculer les termes intégral et dérivé du régulateur PID. La variable error représente l'erreur actuelle entre l'intensité lumineuse moyenne et l'intensité lumineuse mesurée par la troisième photorésistance. Nous calculons également les termes intégral et dérivé en utilisant les erreurs précédentes.

La sortie du régulateur PID est calculée en multipliant chaque terme par son coefficient correspondant et en les sommant. La sortie est ensuite utilisée pour ajuster les angles des servomoteurs de manière à réduire l'erreur et maintenir l'intensité lumineuse moyenne à la valeur souhaitée.

Les conditions de contrôle des servomoteurs sont basées sur la sortie du régulateur PID. Si la sortie est positive, nous augmentons l'angle du servomoteur 1 dans la direction appropriée. Si la sortie est négative, nous réduisons l'angle du servomoteur 1. Les valeurs limites des angles des servomoteurs peuvent être ajustées en fonction des spécifications de notre système.

L'implémentation du régulateur PID nous permet de maintenir l'intensité lumineuse moyenne à la valeur cible en ajustant de manière dynamique les angles des servomoteurs.