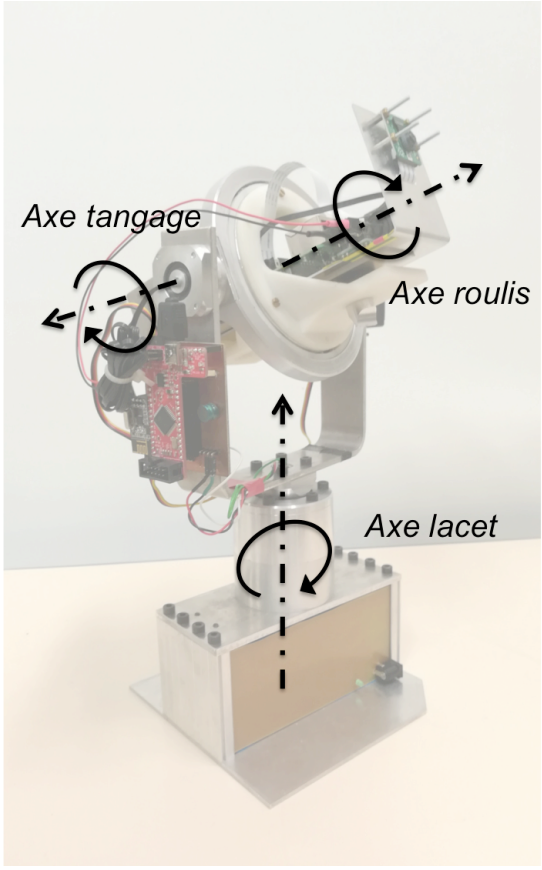
TP PiWebcam

Le système PiWebcam est en système pluri-technique répondant à la problématique suivante.

> *"Piloter une webcam à distance en commandant son orientation et son zoom à l'aide d'un serveur embarqué."*

Une documentation technique associée est disponible, ainsi que la datasheet de la plupart des éléments composants le système.

Nous explorerons durant ce TP le fonctionnement d'un servomoteur usuel et la génération de ses signaux de commande. Nous mettrons en œuvre une PWM hardware en manipulant les registres spécifiques à l'Atmega32U4.



1. Approche globale

<u>Attention :</u> Avant la mise sous tension du système, veuillez lire attentivement la documentation associée et répondre aux questions suivantes.

Q1.1. Établir un tableau récapitulatif des puissances électriques consommés par les différents sous-systèmes de la PiWebcam :

- Servomoteurs (alimenté en 5V, 7V) ;
- Raspberry ;
- microcontrôleur.

Q1.2. Trois adaptateurs secteurs vous sont proposés (1,8W, 30W et 72W). En déduire de la Q1 l'adaptateur le plus approprié pour le bon fonctionnement du système. Présenter et justifier votre résultat à l'enseignant pour qu'il puisse le valider votre choix.

Q1.3. Un condensateur 2200uF a été ajouté en amont de l'alimentation à découpage associée à la Raspberry. Expliciter son rôle ?

Q1.4. Quel type de communication entre la Raspberry et le microcontrôleur est-il mis en œuvre ? D'autres liaisons pourraient-elles être envisagées ?

Q1.5. Les servomoteurs sont-ils commandés en position et/ou vitesse ? Quelles grandeurs physiques les commandent ?

- Tension ;
- Courant ;
- Autres (précisez…)

À présent, alimentez le système avec l'adaptateur retenu et validé à la Q2 et prenez en main le système en vous connectant en Wifi à son intranet.

SSID : PiWebcam
Mot de passe : mdppiwebcam

Attendre deux bonnes minutes que les processus de la Raspberry (mise en route du serveur web, streaming de la webcam…) s'exécutent correctement puis ouvrir un navigateur et taper l'adresse suivante pour accéder à son interface graphique.

192.168.3.1:8081

Manipuler le système via cette interface pour le prendre en main.

Attention : ***L'extinction de la PiWebcam se fait via l'interface graphique. Ce n'est qu'une fois la LED verte de la Raspberry éteinte que vous pouvez retirer l'alimentation.***

2. Commande en position d'un servomoteur

Q2.1. En vous inspirant du programme *.ino* présent dans le dossier */Arduino/config/ArduinoPWM* qui configure le microcontrôleur, écrire un programme simple sous l'IDE Arduino qui permet de piloter la position angulaire de l'arbre de sortie du servomoteur Roulis (pin D10) par largeur d'impulsion. *Vous pourrez vous aider de la bibliothèque Servo d'Arduino.*

Q2.2. À l'aide d'un oscilloscope, vérifier que la PWM générée est bien celle attendue et tracer à l'aide d'Excel un graphe présentant la valeur angulaire de l'arbre de sortie en fonction de la largeur d'impulsion du signal envoyé (la largeur comprise sera entre 700us à 2200us). Quelle est la nature de cette relation ? Existe-t-il des zones mortes ?

Q2.3. Quels phénomènes non désirés remarqués vous sur la commande des servomoteurs ? Ces phénomènes sont particulièrement visibles sur le système PiWebcam dans sa position de repos (position adoptée au démarrage). Justifier à l'aide de l'oscilloscope que ces phénomènes proviennent des signaux PWM envoyés aux servomoteurs.

3. Générations de PWM hardware

Pour supprimer les phénomènes constatés à la Q8, il est nécessaire de générer des signaux PWM matériellement (sans utiliser la bibliothèque Servo d'Arduino mais les ressources matérielles de l'Atmega32U4).

Q3.1. À partir des datasheets disponibles sur le microcontrôleur, expliquer brièvement comment peut-on générer une PWM à partir d'un Timer. Quels types de PWM peut-on générer avec un Atmega32U4?

Q3.2. Quels périodes doivent avoir les signaux PWM ?

Q3.3. Vu les pins sur lesquels sont branchés les servomoteurs de la PiWebcam, quel Timer pouvons nous utiliser ? Quelle est sa résolution (en nombre de bits) ? Sachant que l'on souhaite une résolution angulaire de 0,1°, sera-t-elle suffisante ? Justifier vos affirmations à l'aide des datasheets des servomoteurs.

Q3.4. Quels registres permettent de configurer le Timer ?

Nous souhaitons mettre en œuvre un signal Fast PWM 16bit utilisant le Timer 1 sur le pin D10 du microcontrôleur, afin de commander le Servo HS-5565 assurant la rotation autour de l'axe de roulis.

Q3.5. À l'aide des réponses précédentes, déterminer le prescaler du Timer et sa valeur MAX.

Q3.6. Comparer vos valeurs d'initialisation des registres avec ceux du programme *PWM.ino* disponible dans le dossier *Arduino*. Sont-ils semblables ? Commenter et argumenter vos différences.

Q3.7. Tester le programme et vérifier à l'oscilloscope la PWM généré. Est-elle celle escomptée ?

Q3.8. Le programme MyOwnPWM.ino reprend le programme ArduinoPWM.ino en n'utilisant plus la bibliothèque Servo mais en générant une Fast PWM 16bit comme vu précédemment. Flasher le programme dans le microcontrôleur, retourner sur l'interface graphique de la PiWebcam et piloter le système. Les phonèmes constatés à la Q8 sont-ils toujours présents ? Commenter et justifier vos affirmations.