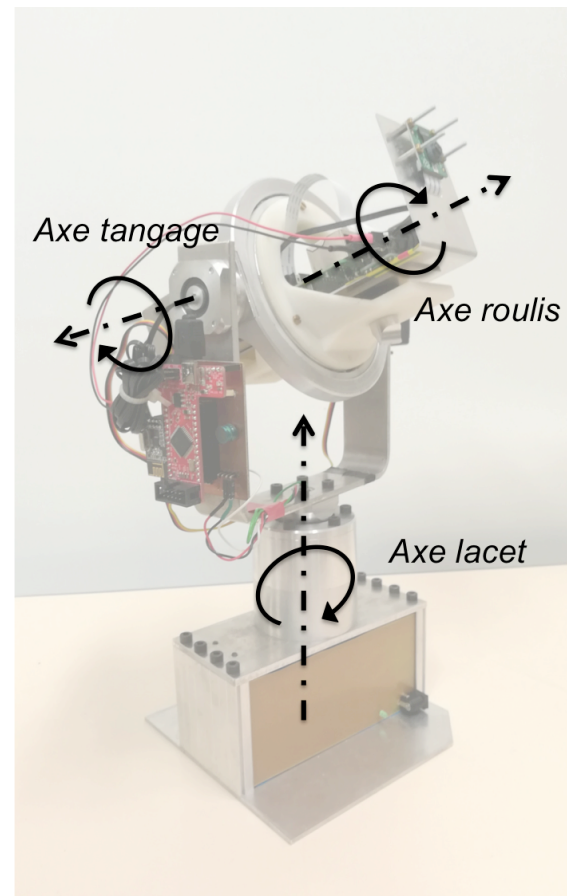


# TP PiWebcam

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

*“Piloter une webcam à distance en commandant sa direction 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 mise en œuvre de ses signaux de commande.



## Approche globale

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

**Q1.** Établir un tableau récapitulatif des puissances électriques consommées par les différents sous-systèmes de la PiWebcam.

**Q2.** Deux adaptateurs secteur vous sont proposés (1,8W 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 valider votre choix.

**Q3.** Un condensateur 2200uF a été ajouté en amont de l’alimentation à découpage associée à la Raspberry. Expliciter son rôle ?

**Q4.** Quel type de communication entre la Raspberry et le microcontrôleur est-il mis en œuvre ? D’autres liaisons pourraient-elles être envisagées ?

**Q5.** Quelles sont les grandeurs physiques qui commandent les servomoteurs ? Les servomoteurs sont-ils commandés en position et/ou vitesse ? Expliquer leur fonctionnement.

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*

Ouvrer un navigateur et taper l'adresse suivante pour accéder à son interface graphique.

*192.168.3.1:8081*

## Commande en position d'un servomoteur

**Q6.** 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 d'un arbre de sortie d'un servomoteur par largeur d'impulsion. Vous pourrez vous aider de la bibliothèque Servo d'Arduino.

**Q7.** 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. Quelle est la nature de cette relation ? Existe-t-il des zones mortes ?

**Q8.** Quels phénomènes non désirés remarquables vous sur la commande des servomoteurs ? Ces phénomènes sont particulièrement visibles sur le système PiWebcam dans sa position de repos (position adoptée au démarrage). D'après vous d'où peuvent-ils venir ?

## Générer des pulses de largeur d'impulsion variable (PWM)

Pour supprimer les phénomènes constatés à la Q8, il est nécessaire de générer des signaux PWM sans utiliser la bibliothèque Servo d'Arduino.

**Q9.** 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 existent-ils ?

**Q10.** Vu les pins sur lesquels sont branchés les servomoteurs de la PiWebcam, quel Timer pouvons nous utiliser ? Quelle est sa résolution ? Sera-t-elle suffisante ? Justifier vos affirmations à l'aide de la datasheet des servomoteurs.

**Q11.** Quelles périodes doivent avoir les signaux PWM ?

**Q12.** Quels registres permettent de configurer le Timer ? Même question concernant les signaux PWM ?

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.

**Q13.** À l'aide des réponses précédentes, déterminer le prescaler du Timer et sa valeur MAX.

**Q14.** Comparer vos résultats avec ceux du programme PWM.ino disponible dans le dossier Arduino. Sont-ils semblables ? Commenter et argumenter vos différences.

**Q15.** Tester le programme et vérifier à l'oscilloscope la PWM généré. Est-elle celle escomptée ?

**Q16.** 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 phénomènes constatés à la Q8 sont-ils toujours présents ? Commenter et justifier vos affirmations.

--

**Fin**