

# Journal de bord MOOREV-Timer-Camera ScFo

## Journal de bord:

**06/03:**

- Pour notre système il faut que l'on choisisse entre l'Arduino NANO et l'ESP32 en se basant sur leur consommation en énergie. Il faut que l'on puisse mesurer la consommation en énergie sur différentes séquences et différentes durée.

**09/03: [séance Fablab]**

- Durant la séance nous avons discuté du projet et en quoi le Fablab peut nous aider. Réflexion sur les tests et les protocoles à réaliser pour la consommation de la batterie avec les deux cartes, avec recherches sur la mise en veille des cartes. Recherches sur les aspects de la GoPro Hero3+ et comment la manipuler avec un programme implanté. Réflexion sur les durées et les séquences d'enregistrements pour la caméra.

- Visite de l'atelier Fablab partie électronique, avec présentation du matériel mis à disposition. Préparation du matériel pour la prochaine séance au Fablab.

**10/03: [séance Fablab]**

- Point et vu d'ensemble sur tout le matériel que nous possédons.

- Point sur la façon de relier le micro processeur à la GoPro. Recherches sur le port Hero de la GoPro Hero 3+ qui nous permet de conclure sur l'utilisation du port Hero pour contrôler de manière filaire la caméra. Recherche de références afin de pouvoir trouver et se procurer un branchement compatible au port Hero.

## Liens:

Autre possibilité pensée de connecter la GoPro qui nécessite la modification de la caméra (idée mise de côté):

<https://hackaday.io/project/12108-gopro-hero-4-wired-control>

<https://www.youtube.com/watch?v=mpPfk4sLb3I>

Informations sur le port Hero Bus de la GoPro Hero:

<https://wiki.logre.eu/index.php/GObotPRO>

### **17/03: [séance Fablab]**

- Réunion afin de préparer les tests de consommation d'énergie entre l'Arduino Nano et l'ESP32. Acquisition du matériel nécessaire, et codage des cartes.
- Tests (code) du nombre de cycles par minutes possible sur l'Arduino qui va permettre d'allumer et d'éteindre la caméra.
- Emprunt d'un ESP32 qu'on a du souder au labo électronique.
- Création d'une page GitHub afin de déposer notre code et recherches sur le mode hibernation de l'esp32 ainsi que la façon de récupérer les données d'un multimètre sur l'ordinateur afin d'enregistrer les données.

### **24/03: [séance Fablab]**

- Prise de mesures des consommations.
- Programmation de l'ESP32 en mode basse consommation, mise à disposition du code sur Github. Réalisation d'un montage sur planche de prototypage pour mesurer la consommation des microcontrôleurs avec un multimètre.

([Lien vers le GitHub](#))

- Téléchargement d'un logiciel lié au multimètre afin de le connecter à ce dernier pour récolter en temps réel les données mesurées.
- Réalisation des tableaux Excel avec les résultats.

### **31/03: [Séance Fablab]**

- Réception du matériel nécessaire (ESP32-CAM et port Hero) pour connecter le microcontrôleur à la GoPro.
- On procède ensuite à la soudure de fils sur le port de chaque pins afin de pouvoir accéder aux pins
- Réflexion sur la programmation de l'ESP32-Cam
- Calcul sur tableau Excel des différentes séquences possible en prenant en compte la capacité de stockage de la carte SD et de la taille d'une vidéo/photo.

Tableau de séquence dans le répertoire GitHub: [Lien vers le GitHub](#)

#### **04/04: [Séance Fablab]**

- Continuation de la soudure des pins sur le connecteur
- Programmation de l'ESP32 Cam et test des photos. Mesure de leur capacité de stockage photos et vidéos. Rencontre d'un problème: ESP32 cam -> focal fixe par défaut avec le capteur OV2640, à remplacer par un autre capteur plus adapté ([lien vers le capteur](#)).
- Test des pins du connecteur et confirmations des ports (port 19 pour allumage de la caméra). Test lancer un script sur carte SD en allumant la GoPro.

#### **07/04: [Séance Fablab]**

- Nous avons constaté que la soudure n'a pas tenue et que des broches ont été cassé lors de la manipulation. Nous réfléchissons donc à un autre moyen de souder des fils moins rigides, ou bien à une alternative.
- Le but de la séance était d'allumer la GoPro à l'aide de l'arduino et du port Hero, et de vérifier que le script se lançait à l'allumage de la caméra. Nous rencontrons toutefois un problème concernant la lecture du script, que la caméra ne détecte pas.

#### **14/04: [Séance Fablab]**

- Tests du fonctionnement du script sur la GoPro. Préparation de la présentation à l'oral et des comptes rendus concernant l'UE.

Le script lancé est disponible sur le répertoire GitHub sous le nom de 'autoexec.ash'.

([Lien vers le GitHub](#))

(Une fois que nous pouvons connecter l'Arduino à la GoPro, nous devons nous assurer que la GoPro lit bien le script présent sur la carte SD dès l'allumage. Pour cela, nous allons déposer dans la racine de la carte un fichier sous le nom de 'autoexec.ash' et qui contient les instructions. )

#### **18/04: [Séance Fablab]**

Dernière séance avant le rendu des travaux. Nous avons testé une dernière fois le lancement du script sur la GoPro en branchant le tout sur l'Arduino. Nous avons fait face à quelques problème au niveau des branchements qui n'allumait pas la caméra. Le problème a été résolu à la fin grâce à l'aide d'un chercheur présent au Fablab.

Le prototype fonctionne, il ne reste qu'à programmer la bonne séquence sur le script de la GoPro.