Table des matières

On se documente ? 3
On réfléchit... 3
On prépare ! 3
On complète ! 4
On comprend !!! 5

On se documente ?

- Résolution vidéo : 720×480
- Fonction audio : Oui
- Image/sec : 28fps
- Taille de la tête pivotante : 25x25x28m
- Led : Bi-colore
- Type de carte : Micro SD-Card
- Rayon : 0.3 à l'infini
- Tension : 4.5V-6V
- Consommation : 180Mah
- Objectif : Orientable à 90°
- USB : Mini USB 1.1
- Modes : Vidéo, déclenchement à distance, photo enregistrement à 180° programmable

http://tic.technologiescollege.fr/wiki/doku.php/fr/arduino/flycam

http://www.camonetec.com/de/produkte/kamerasysteme/flycamone-eco-v2.html

Donc il faudra lui envoyer des ***commandes de servomoteur***, ça tombe bien des blocs sont déjà prêts…

LECOPIER-COLLERVAETRENOTREMEILLEURAMI!

On réfléchit…

Au final il n'y aura que 3 fonctions :

- switcher le mode vidéo / photo / photos en série : 180° pendant 3s
- commencer une vidéo / prendre une photo / déclencher une série de photos : impulsion servomoteur de 180°
- arrêter : impulsion servomoteur de 180°

Là où on teste, on s'aperçoit qu'il faudra aussi rajouter une impulsion servomoteur de 0° comme si on relâchait le bouton de commande.

Donc en **entrée** nous n'avons besoin que de la **broche numérique** compatible avec des commandes de servomoteur.

On prépare !

Dans le dossier '**\blocks**', il faut créer un dossier avec le nom du projet ou fabricant, un fichier de *texte brut avec le même nom* et toutes les images nécessaires à illustrer les blocs :

-> le nom du dossier est unique et explicite

-> le nom du dossier est celui du fichier js

-> le nom de l'image sera le nom de la fonction

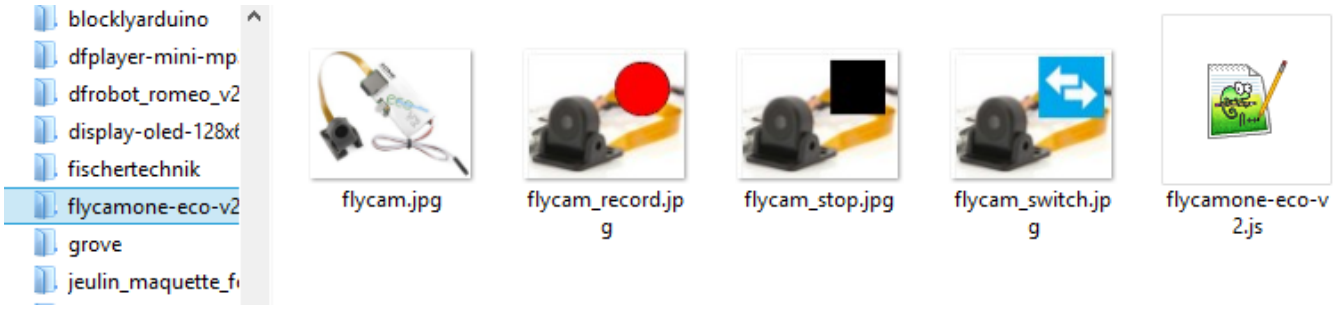
-> pas de majuscules !

-> les espaces et accents sont interdits !!!

→ *de préférence tout en Anglais, langue de base en programmation pour ensuite faire des traductions…*

→ *ne pas avoir peur des noms longs, il faut que ce soit explicite !*

C'est un peu comme les 10 commandements, mais moitié moins…les deux dernières chacun en fait ce qu'il veut mais c'est mieux.



On complète !

Dans le fichier javascript '**flycamone-eco-v2.js**' il faut décrire le bloc. En regardant la documentation de Google Blockly, ainsi que les blocs existants (***modifiez un paramètre et regardez ce que ça produit comme changement***) notamment les servomoteur dans notre cas.

```
/**
 * Block pour la FlycamOne Eco v2
 * @author Seb Canet (canet.s@free.fr)
 */
'use strict';

goog.provide('Blockly.Blocks.flycam');
```

```
goog.require('Blockly.Blocks');
goog.require('Blockly.Types');

Blockly.Blocks.flycam.HUE = '#46C286';

Blockly.Blocks.flycam_switch = {
  init: function() {
    this.setColour(Blockly.Blocks.flycam.HUE);
    this.setHelpUrl(Blockly.Msg.FLYCAM_SWITCH_HELPURL);
    this.appendDummyInput("")
        .appendField(Blockly.Msg.FLYCAM_SWITCH_TEXT)
        .appendField(new Blockly.FieldImage(Blockly.pathToBlockly +
'blocks/flycamone-eco-v2/flycam_switch.jpg', Blockly.Arduino.imageSize,
Blockly.Arduino.imageSize));
    this.appendValueInput("PIN")
        .setCheck('Number')
        .setAlign(Blockly.ALIGN_RIGHT)
        .appendField(Blockly.Msg.FLYCAM_SWITCH_INPUT);
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setTooltip(Blockly.Msg.FLYCAM_SWITCH_TOOLTIP);
  }
};
```

On comprend !!!

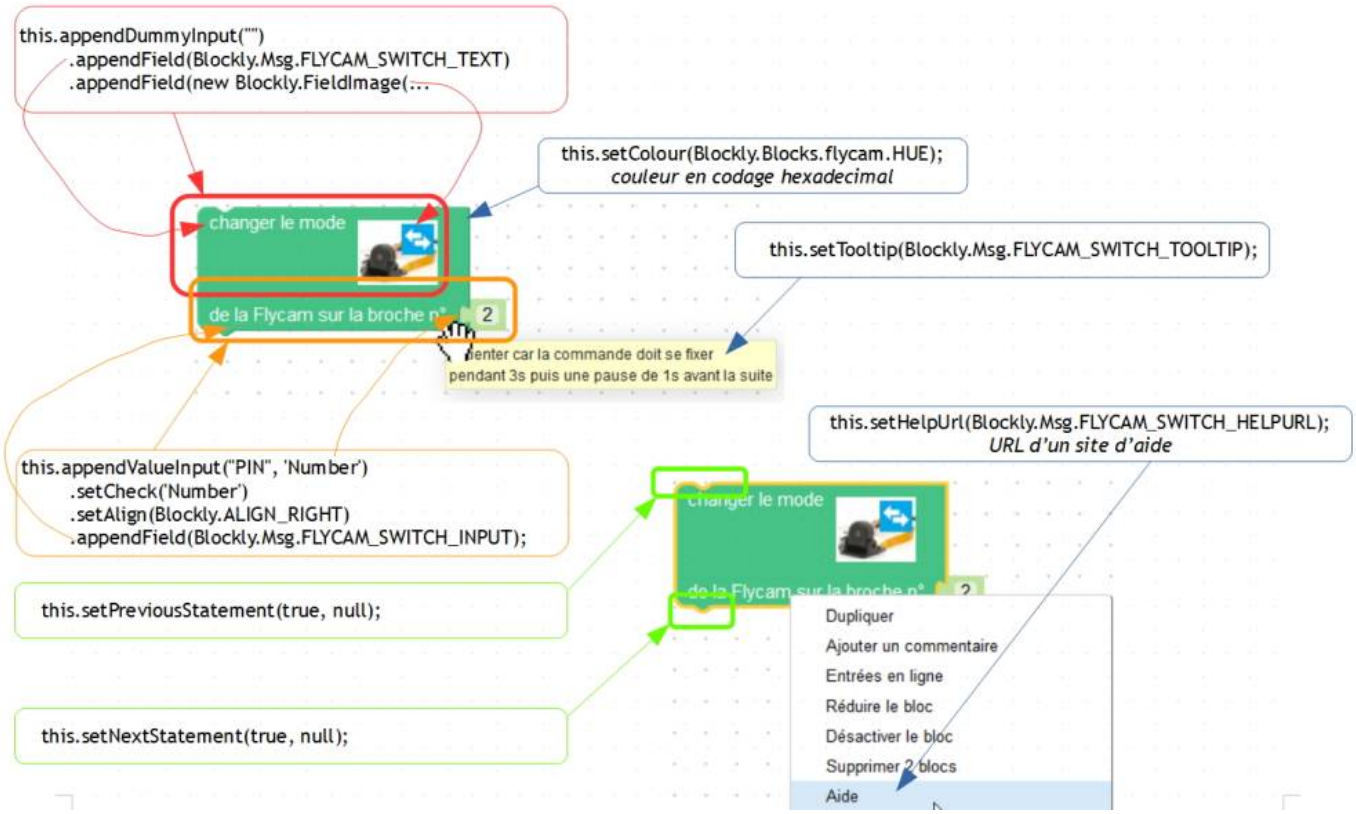
Décryptons tout ça dans l'ordre :

1. tout ce qui est entre **/*** jusqu'à ***/** est considéré comme un commentaire, il est toujours bon d'être explicite
2. la première ligne est nécessaire
3. ce fichier va donc fournir des blocs de la bibliothèque 'flycam' ou ce que vous voulez, d'où le : goog.**provide** ('Blockly.Blocks.**flycam**');
4. mais on a besoin de la bibliothèque des formes définie par Blockly **:** goog.**require** ('Blockly.Blocks');
5. on définit la couleur de tous les blocs : Blockly.Blocks.**flycam**.HUE= '#46C286';
    1. **/!\ ATTENTION /!** je l'affiche ici pour compréhension : la variable Blockly.Blocks.flycam.HUEva contenir la valeur '#46C286', **mais il faut maintenant la déplacer** dans le fichier **'\blocks\blocks_colors.js'**. Il suffira de l'utiliser par la suite
6. c'est parti pour le nom du 1er bloc : *Blockly.Blocks.flycam_switch* (**sans espace ni accent, qui porte le nom de la photo qui va avec, etc**)
7. on ouvre la définition de la fonction : init: function() {
8. rapido on attribue la couleur de **ce** bloc : **this**.setColour(Blockly.Blocks.**flycam**.HUE); l'intérêt est d'utiliser la variable en n'ayant besoin de lui attribuer une valeur qu'une seule fois
9. pour que les gens disposent d'une aide au clic droit : this.setHelpUrl(Blockly.Msg.FLYCAM_SWITCH_HELPURL); **/!\ ATTENTION /!** on n'utilise QUE des variables : la variable Blockly.Msg.FLYCAM_SWITCH_HELPURLva contenir l'adresse du site web, on verra comment le remplir par la suite
10. on fait une **entrée** standard pour la remplir avec des infos : this.appendDummy**Input** ("")…

11. ...qui contiendra un texte contenu dans une variable: .appendField(Blockly.Msg.FLYCAM_SWITCH_TEXT)...
12. ..avec ensuite une photo dont les dimensions vont changer à la volée (donc des variablespour la taille !) : .appendField(new Blockly.FieldImage(Blockly.pathToBlockly + 'blocks/flycamone-eco-v2/flycam_switch.jpg', Blockly.Arduino.imageSize, Blockly.Arduino.imageSize));
13. nouvelle **entrée** qui s'affichera donc dans le bloc sur une autre ligne, mais cette fois ce n'est pas une entrée pré-remplie mais qui est ouverte à une *valeur extérieur* de type 'Number' et stockée dans la variable 'PIN'(hyper importante celle-là car elle servira au code Arduino !!!) : this.appendValue**Input** ("PIN")...
14. ...on **vérifie** que le bloc rajouté sera un *chiffre* (et non pas un texte, etc) : .set**Check** ('*Number*')...
15. ...et présentée par un court texte contenu dans une variable: .appendField(Blockly.Msg.FLYCAM_SWITCH_INPUT);
16. on stipule que ce bloc *peut* s'emboîter avec d'autres **avant lui** : this.set**Previous**Statement(*true*, null);
17. on stipule que ce bloc *peut* s'emboîter avec d'autres **après lui** : this.set**Next**Statement(*true*, null);
18. on propose au survol de la souris qu'apparaisse un ballon avec de laide contenue dans la variable: this.setTooltip(Blockly.Msg.FLYCAM_SWITCH_TOOLTIP);
19. on referme tout ça avec des accolades et des points virgules, vous verrez c'est pénible à vérifier mais il faut à tout prix que ce soit équilibré !

Donc il faut bien préparer ses blocs en amont, sinon n'aurait pas su qu'il fallait inventer :

- 1 nom de fonction,
- 6 variables + 1 imposée (Blockly.Arduino.imageSize)
- 1 entrée
- 4 textes



From:
https://wiki.libreduc.cc/ - **LibrEduc**

Permanent link:
https://wiki.libreduc.cc/fr:arduino:blockly_rduino:creerblocsmultiling:blocdef

Last update: **2025/01/16 20:24**