 **Warning:** Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**



SMART BUD

Demo (<https://bud.eco-sensors.ch>) | Tutoriels (<https://eco-sensors.ch/tutoriels/>)



NOUS SOUTENIR

Shop (<https://eco-sensors.ch/shop/>) & Sponsoring (<https://eco-sensors.ch/product-category/dons/>)



SOCIAL

Facebook (<https://www.facebook.com/ecosensors>) - Contact (<https://eco-sensors.ch/contact/>)

Snap! Si tu bouges, je t'immortalise!

HOME (HTTPS://ECO-SENSORS.CH/)
/ SYSTÈMES EMBARQUÉS (HTTPS://ECO-SENSORS.CH/
CATEGORY/SYSTEMES-EMBARQUES/)
/ SNAP! SI TU BOUGES, JE T'IMMORTALISE!

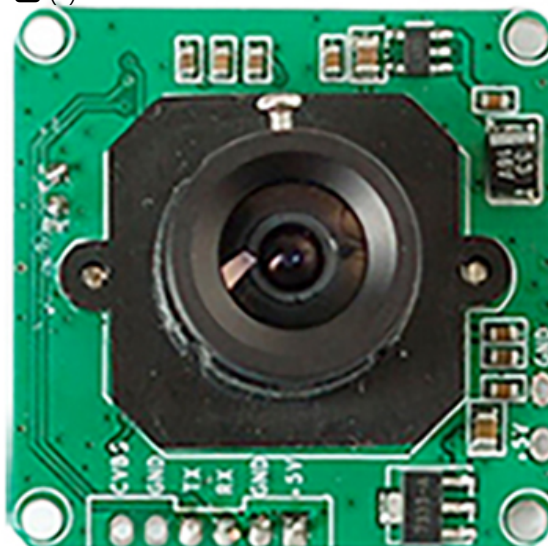
BY ECOSENSORS (HTTPS://ECO-SENSORS.CH/AUTHOR/
ECOSENSORS/)

/

23 DÉCEMBRE 2017 (HTTPS://ECO-SENSORS.CH/SNAP-SI-TU-
BOUGES-JE-TIMMORTALISE/)

 0  2,694  0  

SYSTÈMES EMBARQUÉS (HTTPS://ECO-SENSORS.CH/CATEGORY/
SYSTEMES-EMBARQUES/)



Dans ce petit exercice, je vais résumer comment connecter une camera pour que lorsque votre chat passe devant un détecteur de mouvement, la camera l'immortalise.

Vous allez apprendre comment détecter un mouvement avec un capteur de mouvement PIR (<https://learn.adafruit.com/active-infrared-proximity-motion-sensor/how-pirs-work>) et **sans** détecteur de mouvement!

Plus précisément, sans le capteur de mouvement PIR, vous allez configurer votre camera pour qu'elle compare les pixels de deux images. C'est cette différence qui va déclencher la capture de l'image.

Contrairement à cet article « transformer votre raspberry en camera de surveillance (<https://eco-sensors.ch/3-transformer-votre-raspberry-en-camera-de-surveillance/>) » nous n'allons **pas** utiliser un Raspberry, mais un micro-contrôleur Adafruit Feather MO Adalogger (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/overview>) avec un processeur ATSAMD21G18 ARM Cortex M0 (48 MHz) et 256KB de FLASH + 32KB de RAM. Ce micro-contrôleur a un lecteur de carte SD intégré qui nous permettra d'enregistrer les images.

Matériel

- Micro-controlleur avec carte lecteur SD (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/overview>)
- Carte microSD
- Camera TTL Serie 5V (<https://learn.adafruit.com/ttl-serial-camera/circuitpython-usage?view=all>)
- Détecteur de mouvement (PIR) (<https://www.seeedstudio.com/Grove-mini-PIR-motion-sensor-p-2930.html>)
- Librairie Adafruit_VC0706 (<https://github.com/adafruit/Adafruit-VC0706-Serial-Camera-Library>)

Lecture

How PIRs Work from Adafruit (<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>)

Connexions

Pin de la Camera => boche (pin) du micro-controlleur

Vcc => 5V

GND => GND

Rx => Tx

Tx => Rx


Fils du capteur de mouvement => broche micro-controlleur

Vcc => 3.3V

GND => GND

SIG => 12

Code

Tout d'abord, vous devez importer la librairie d'Adafruit VC0706
( <https://github.com/adafruit/Adafruit-VC0706-Serial-Camera-Library>) ainsi que les librairie SPI et SD (dans votre nouveau fichier.ino

```
1 #include <Adafruit_VC0706.h> // https://githu
2 #include <SPI.h>
3 #include <SD.h>
```

Vous devrez encore donner un nom à deux constantes (<https://www.arduino.cc/reference/en/language/structure/further-syntax/define/>) ainsi que sa valeur, qui sera nécessaire au capteur de mouvement. J'attire votre attention que je crée aussi une constante CAMERA_MOTION qui nous permettra d'activer cette fonction ou de la désactiver en la commentant. C'est ceci qui nous permettra de détecter un mouvement sans le détecteur PIR. Vous devez faire de même pour la carte SD

```
1 /*
2  * Motion
3  */
4 #define PIR_PIN 12 // Pin definition for the :
5 #define LED 13 // Pin definition for the LED
6 // #define CAMERA_MOTION // To activate or not
7
8 /*
9  * SD
10 */
11 #define chipSelect 4
```

Si vous utilisez un autre micro-controlleur et une lecteur SD séparé (<https://www.adafruit.com/product/254>), vous devez connecter la broche (pin) CS du lecteur SD à la broche de votre micro-controlleur. Vous pouvez très bien choisir une autre broche que celle que j'ai choisie. Étant donné que le lecteur SD est intégré au micro-controlleur (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/pinouts#micro-sd-card-plus-green-led>) que je propose, la broche 4 est imposée par le constructeur.

Le micro-controlleur a deux port Hardware: **Serial** et **Serial1**.
J'utilise le Serial pour afficher des messages dans le terminal

```
1 Serial.print(F("Hello, voici la valeur de "))  
2 Serial.println(ma_variable);
```

Alors que Serial1 est utilisé pour les ports Rx et Tx.

C'est ces deux broches que nous allons utiliser (rx/Tx) pour communiquer avec la camera. Autrement dit, c'est sur ce port (Hardware) que le micro-controlleur va traiter les données envoyées par la camera. Vous devez donc ajouter ceci à la suite de votre code, qui va créer un objet en indiquant le port à utiliser.

La fonction setup()



CODE QUI SUIT, DOIT SE TROUVER SANS LE
CONTION SETUP()

setup() (<https://playground.arduino.cc/French/Setup>) initialise et fixe les valeurs de démarrage du programme. Il va définir l'état des broches, et initier certaines fonctionnalités.

```
1 Serial.begin(115200);
2 Serial1.begin(115200);
3
4 //define CAMERA_MOTION
5
6 pinMode(PIR_PIN,INPUT_PULLUP); // Define the PIR
7 digitalWrite(PIR_PIN,LOW); // Put to LOW pinMode
8
9 pinMode(LED,OUTPUT); // Define as Input
10 digitalWrite(LED,LOW); // Put to low
```

Vous devez ensuite, initier votre lecteur de carte SD et la camera

```
1 if (!SD.begin(chipSelect)) // see if the card
2 {
3   Si.println(F("Card failed, or not present"));
4   // don't do anything more:
5   while(1); // Do not continue
6 }
7
8 if (cam.begin()){ // Try to locate the camera
9   Si.println(F("Camera Found:"),0);
10 }
11 else
12 {
13   Si.println(F("No camera found?"),0);
14   while(1); // No, Do not continue
15 }
16 /*
17 * Get the version of the Camera
18 */
19 char *reply = cam.getVersion(); // Print out
20 if (reply == 0) {
21   Si.print(F("Failed to get version"),0);
22 }
23 else
24 {
25   Si.print(F("Cam version: "),0);
26   Si.println(reply,0);
27 }
```

Vous pouvez encore définir la taille de vos photos. Plus elles seront lourdes, plus le programme mettra du temps pour les sauver (env. 4 à 24s. pour une taille moyenne)

```
1 // Set the picture size – you can choose one of
2 // Remember that bigger pictures take longer
3
4 //cam.setImageSize(VC0706_640x480); // biggest
5 cam.setImageSize(VC0706_320x240); // medium
6 //cam.setImageSize(VC0706_160x120); // small
7
8 // You can read the size back from the camera
9 uint8_t imgsize = cam.getImageSize();
10 Serial.print(F("Image size: "));
11 if (imgsize == VC0706_640x480) Serial.println("640x480");
12 if (imgsize == VC0706_320x240) Serial.println("320x240");
13 if (imgsize == VC0706_160x120) Serial.println("160x120");
```

Finalement, nous allons ici spécifier si nous utilisons la détection avec le détecteur PIR ou si la camera va elle remplir cette fonction.

```
1 #ifdef CAMERA_MOTION // If defined or commen
2 // Motion detection system can alert you whe
3 cam.setMotionDetect(true); // turn it on
4 #else
5 cam.setMotionDetect(false); // turn it off (
6 #endif
7 // You can also verify whether motion detect:
8 Serial.print("Camera Motion detection is ");
9 if (cam.getMotionDetect())
10 Serial.println("ON");
11 else
12 {
13 Serial.println("OFF");
14 Serial.println("PIR sensor ON");
15 snap(); // Take a snap (Can be commented)
16 }
```

La fonction loop()

**❗ TOUT LE CODE QUI SUIV, DOIT SE TROUVER SANS LE
CONTION LOOP()**

loop() (<https://playground.arduino.cc/French/loop>) (boucle en anglais) fait exactement ce que son nom suggère et s'exécute en boucle sans fin, permettant à votre programme de s'exécuter et de répondre.

Pour que la détection de mouvement se fasse grâce à la camera, soit sans le détecteur PIR, vous devez dé-commenter ceci

```
1 // #Define CAMERA_MOTION
```

Ainsi, le code fera appelle à la fonction *motionDetected()* qui retournera TRUE si la camera détecte un mouvement. La détection des mouvements sera temporairement désactivée le temps de prendre la photo et de la sauver sur la carte SD.

```
1 void loop(){
```

```

2  #ifdef CAMERA_MOTION // IF camera is set to :
3  if (cam.motionDetected()) // If the camera de
4  {
5  Serial.println("Motion!");
6  cam.setMotionDetect(false); // Turn off the
7  snap(); // Take a picture and save it
8  cam.resumeVideo();
9  cam.setMotionDetect(true); // Turn on the fu
10 }
11 #else
12 if(isPIRMotionDetected()==true) // Call the
13 { // will return true if the PIR sensor dete
14 Serial.println("PIR!");
15 snap(); // take a picture and save it
16 }
17 #endif
18 }

```

Si en revanche, vous n'activez pas la détection par la camera, la fonction *isPIRMotionDetected()* sera appelée et le détecteur PIR indiquera au micro-contrôleur si un mouvement est détecté ou pas.

Avec le détecteur PIR

Vous pouvez commencer par un peu de lecture (en anglais) sur les détecteurs de mouvements PIR (<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>).

Le fil SIG est donc connecté à la broche 12 (pin 12) qui elle, attend (INPUT) un état haut ou bas. Quand un mouvement est détecté par le détecteur PIR, la broche 12 passe à l'état haut. *isPIRMotionDetected()* retourne TRUE et la fonction *snap()* va lancer le processus de la prise de vue et sauvegarder la photo sur la carte SD.

isPIRMotionDetected()

```
1 boolean isPIRMotionDetected()
2 {
3   int sensorValue = digitalRead(PIR_PIN); // Read the sensor value
4   //Serial.print("SensorValue:"); Serial.print(sensorValue);
5
6   if(sensorValue == HIGH) // if the sensor value is HIGH
7   {
8     Si.println(F("\r\nMotion detected!!"),0);
9     digitalWrite(PIR_LED,HIGH);
10    delay(500);
11    digitalWrite(PIR_LED,LOW);
12    return true; // yes,return true
13  }
14  else
15  {
16    return false; // no,return false
17  }
18 }
```


La camera

Comme indiqué plus haut, quand *isPIRMotionDetected()* ou si *cam.motionDetected()* retournent TRUE, la fonction *snap()* est appelée.

```
1 void snap()
2 {
3   if (! cam.takePicture())
4   {
5     Si.sprintln(F("Failed to snap!"),0);
6   }
7   else
8   {
9     Si.sprintln(F("Picture taken!"),0);
10
11 // Create an image with the name IMAGExx.JPG
12 // Increment the file name if it exists
13 char filename[13];
```

```

14 strcpy(filename, "IMAGE00.JPG");
15 for (int i = 0; i < 100; i++) {
16     filename[5] = '0' + i/10;
17     filename[6] = '0' + i%10;
18     // create if does not exist, do not open exist
19     if (! SD.exists(filename)) {
20         break;
21     }
22 }
23
24 File imgFile = SD.open(filename, FILE_WRITE);
25 uint16_t jpglen = cam.frameLength(); // Get
26 Si.sprintf(F("Storing "),0);
27 Si.sprintf(jpglen, DEC,0);
28 Si.sprintf(F(" byte image in "),0);
29 Si.println(filename,0);
30
31 int32_t time = millis(); // Init the time for
32
33 byte wCount = 0; // Read all the data up to
34 while (jpglen > 0) // For counting # of write
35 {
36     uint8_t *buffer;
37     uint8_t bytesToRead = min(64 , jpglen); // cl
38     buffer = cam.readPicture(bytesToRead);
39     imgFile.write(buffer, bytesToRead);

```

La fonction ci-dessus est relativement simple à comprendre

La photo est prise grâce à

```

1 cam.takePicture();

```

En suite, la boucle *for* va parcourir votre carte SD et va contrôler les fichiers commençant par IMAGE00.JPG pour donner un nom à la nouvelle image qui finira par un numéro, pour autant que le nom n'existe pas encore. Si par exemple :

IMAGE00.JPG

IMAGE01.JPG

IMAGE03.JPG

La nouvelle image sera IMAGE02.JPG et la suivante IMAGE04.JPG

 (/)   (/)
(etc).

La fonction

```
1 SD.open(filename, FILE_WRITE)
```

va ouvrir le nouveau fichier en mode écriture.

La boucle

```
1 while (jpglen > 0){}
```

va « écrire » ou sauver la photo prise dans le fichier .

La fonction

```
1 imgFile.close();
```

va « fermer » le fichier.

Bravo!

Vous avez mis en place un petit système espion pour capturer un intru. Vous pouvez aussi vous passer du capteur de mouvement afin de configurer votre caméra pour qu'elle détecte elle-même un mouvement.

Si vous avez aimé cet article, publiez-le!

📌 **ADAFRUIT** ([HTTPS://ECO-SENSORS.CH/TAG/ADAFRUIT/](https://eco-sensors.ch/tag/adafruit/))
ADALOGGER ([HTTPS://ECO-SENSORS.CH/TAG/ADALOGGER/](https://eco-sensors.ch/tag/adalogger/))
ATSAMD21G28 ([HTTPS://ECO-SENSORS.CH/TAG/ATSAMD21G28/](https://eco-sensors.ch/tag/atsamd21g28/))
CORTEX ([HTTPS://ECO-SENSORS.CH/TAG/CORTEX/](https://eco-sensors.ch/tag/cortex/)) **PIR** ([HTTPS://ECO-SENSORS.CH/TAG/PIR/](https://eco-sensors.ch/tag/pir/))

PREV POST

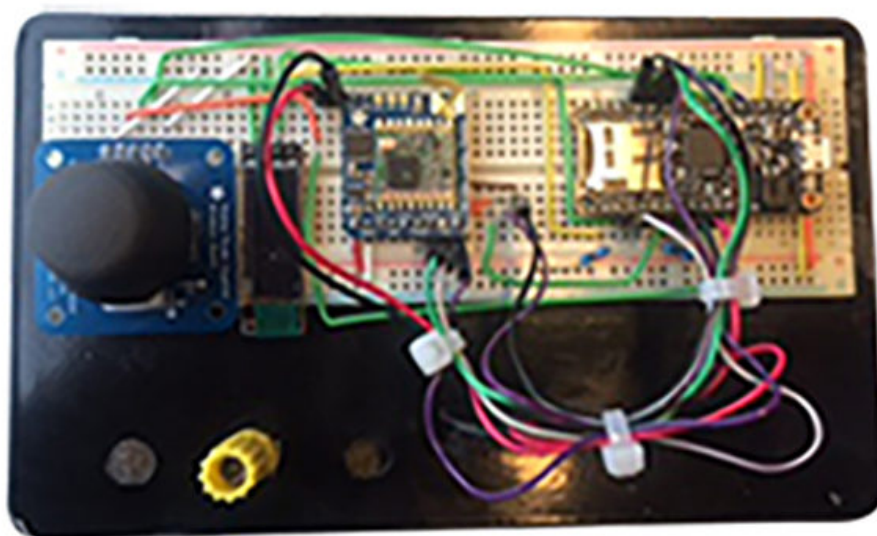
NEXT POST

(<https://eco-sensors.ch/un-joystick-pour-orienter-un-petit-robot-avec-lora/>)

(<https://eco-sensors.ch/un-raspberry-zero-un-lcd-et-lora-pour-recevoir-des-donnees-rfm95/>)

Mesure de la qualité de l'air (PM2.5/PM10) –

15 AVRIL 2020



Un Joystick pour orienter un petit robot avec


18 DÉCEMBRE 2017

Leave **Comment:**


FULL NAME

EMAIL ADDRESS


PHONE NUMBER



YOUR COMMENT



YOUR COMMENT



YOUR COMMENT

SUBMIT NOW

