



Découvrez notre Chaîne YouTube "[Ingénierie et Projets](#)"

Découvrez notre Chaîne Secondaire "[Information Neuronale et l'Ingénierie du Cerveau](#)"

Objectifs

- Savoir établir la liaison avec le port série
- Savoir établir la liaison avec le port caméra
- Savoir détecter les mouvements dans une [image](#)
- Savoir transférer les données vers la [carte Arduino](#)
- Savoir commander des LEDs avec [Matlab](#) via la carte Arduino
- S'initier à la commande avec Matlab
- Etc.

Ouverture de la Caméra (Multi-sources)

La fonction `openCam()` permet l'ouverture d'une ou plusieurs caméras (ou source). La source 1 est la caméra de l'ordinateur par défaut. Vous pouvez aussi ouvrir une ou plusieurs caméras branchées dans le port USB, etc. On verra dans le tuto l'utilisation des sources 1 (caméra du PC portable) et 2 (caméra branchée via le port USB). Le paramètre `Type` permet de choisir le type de l'image acquise : 0 (image noir et blanc), valeur non nulle (image couleur). Ci-dessous un exemple de capture d'image et ouverture d'une caméra.

```
Type=0;      % 0(GRAY), 1(RGB)
Source=1;    % 1(CAM1), 2(CAM2), ...
cam=openCam(Type, Source);
```



```
figure(1); imshow(getsnapshot(cam)); return;
```

Définition de la fonction openCam()

```
function cam = openCam(Type, Source)

if Type==0

cam = imaqfind; % Trouver l'objet en mémoire (s'il existe!)

if(~exist('cam'))

cam=videoinput('winvideo',Source);      % Déclaration

set(cam,'ReturnedColorSpace','GRAY');  % GRAY

triggerconfig(cam, 'manual');          % Définition du Trigger

start(cam);                            % Déclenchement

else

delete(cam);

cam=videoinput('winvideo',Source);

set(cam,'ReturnedColorSpace','GRAY');

triggerconfig(cam, 'manual');

start(cam);

end;

end;
```



Projet ARDUINO + MATLAB: Détection de Mouvement - Partie 1/2
[PROJET COMPLET]

```
if Type~=0

cam = imaqfind; % Trouver l'objet en mémoire (s'il existe!)

if(~exist('cam'))

cam=videoinput('winvideo',Source);      % Déclaration

set(cam,'ReturnedColorSpace','RGB');    % RGB

triggerconfig(cam, 'manual');           % Définition du Trigger

start(cam);                             % Déclenchement

else

delete(cam);

cam=videoinput('winvideo',Source);

set(cam,'ReturnedColorSpace','RGB');

triggerconfig(cam, 'manual');

start(cam);

end;

end;

end
```



Ouverture du port série

La fonction `openSerial()` permet l'ouverture du port série. Il faut spécifier le nom du port ainsi la vitesse de transmission. Le transfert est effectué au format 8 bits. Le nom du port doit être identique à celui utilisé par la carte Arduino ! La fonction retourne une erreur à l'absence du port série. Assurez que la carte Arduino est bien branché ainsi l'apparition du nom du port dans l'interface Arduino.

```
namePort='COM3';  
  
baudValue=9600;  
  
SerialCOM = openSerial(namePort, baudValue);
```

Définition de la fonction `openSerial ()`

```
function SerialCOM = openSerial(namePort, baudValue)  
  
% Paramètres de la liaison série (COM)  
  
NumBits=8;           % Nombre de bits  
  
SerialCOM=instrfind;  
  
if(~exist('SerialCOM'))  
  
% Création d'un objet Serial Port  
  
SerialCOM = serial(namePort,'BaudRate',baudValue,'DataBits', NumBits, 'Parity',
```



```
'none');  
  
SerialCOM.Terminator = 'LF';  
set(SerialCOM, 'Timeout',2);  
  
% Connexion du port  
fopen(SerialCOM);  
else  
% Fermeture du port  
delete(SerialCOM);  
  
% Création d'un objet Serial Port  
  
SerialCOM = serial(namePort,'BaudRate',baudValue,'DataBits', NumBits, 'Parity',  
'none');  
  
SerialCOM.Terminator = 'LF';  
set(SerialCOM, 'Timeout',2);  
  
% Connexion du port  
fopen(SerialCOM);  
end;
```



```
end
```

Paramètres du programme

```
M=512; N=512;  
  
Nfil=3; % Taille du Buffer  
  
im_A=zeros(Nfil,M,N);  
  
j=1;  
  
Seuil=0.05;
```

Détection & Transfert vers Arduino

Voir le tuto pour plus de détails

```
while 1  
%% Lecture de l'image courante  
im=imadjust(im2double(getsnapshot(cam)));  
im_in=imresize(im,[M N]);  
  
%% Calcul de l'intensité / Détection  
Moy=squeeze(mean(im_A)) ;  
I_diff=Moy(:)-im_in(:);  
RMSE=sqrt(mean(I_diff.^2));  
DET=10*double(RMSE>Seuil)  
  
%% Mise à jour du buffer  
im_A(j,:,:)=im_in;
```



```
j=j+1;
if j==Nfil+1
j=1;
end;

%% Transfert vers la carte Arduino
if DET~=0
fprintf(SerialCOM,'%d\n',DET);
fprintf(SerialCOM,'%d\n',DET);
fprintf(SerialCOM,'%d\n',DET);
end

%% Image Binaire
Imb=(Moy-im_in)>0.5;

%% Affichage
Im=[Imb im_in];
figure(1); imshow(Im);
end
```

Programme Complet

```
close all; clc; clear all;

%% Ouverture de la Caméra (Multi-sources)
Type=0;      % 0(GRAY), 1(RGB)
Source=1;    % 1(CAM1), 2(CAM2), ...
cam=openCam(Type,Source);
% figure(1); imshow(getsnapshot(cam)); return;

%% Ouverture du port série
namePort='COM3';
baudValue=9600;
SerialCOM = openSerial(namePort, baudValue);

%% Paramètres du programme
M=512; N=512;
Nfil=3; % Taille du Buffer
```



```
im_A=zeros(Nfil,M,N);
j=1;
Seuil=0.05;

%% Détection & Transfert vers Arduino
while 1
%% Lecture de l'image courante
im=imadjust(im2double(getsnapshot(cam)));
im_in=imresize(im,[M N]);

%% Calcul de l'intensité / Détection
Moy=squeeze(mean(im_A)) ;
I_diff=Moy(:)-im_in(:);
RMSE=sqrt(mean(I_diff.^2));
DET=10*double(RMSE>Seuil)

%% Mise à jour du buffer
im_A(j,:,:)=im_in;
j=j+1;
if j==Nfil+1
j=1;
end;

%% Transfert vers la carte Arduino
if DET~=0
fprintf(SerialCOM,'%d\n',DET);
fprintf(SerialCOM,'%d\n',DET);
fprintf(SerialCOM,'%d\n',DET);
end

%% Image Binaire
Imb=(Moy-im_in)>0.5;

%% Affichage
Im=[Imb im_in];
figure(1); imshow(Im);
end
```

[Accueil Traitement d'Images | Matlab](#)

Click to rate this post!



Projet ARDUINO + MATLAB: Détection de Mouvement - Partie 1/2
[PROJET COMPLET]

[Total: 0 Average: 0]

Nous Soutenir