



Sommaire

- [1 Objectifs](#)
- [2 Principe](#)
- [3 Le programme Matlab](#)
- [4 Le programme Arduino](#)

Objectifs

- Savoir détecter un objet en fonction de la couleur
- Comprendre le principe de détection et classification par couleurs
- Savoir générer une action avec [Arduino](#) au moment de la détection
- Savoir transférer l'état du détecteur à la [carte](#) Arduino
- Savoir coder les couleurs sur 1 bit ou N bits
- Etc.

Principe

Le principe consiste à commander trois LEDs (sorties TOR) en fonction de la valeur acquise via le port série. Autrement dit, en fonction de la couleur détectée. Les LEDs sont branchées dans les pins 50, 48 et 46 de la carte Arduino Mega. La fonction `Serial.parseInt()` permet de scruter le port série à la recherche d'une valeur entière, si la valeur est égale à :

- R=10 : Allumage de la [LED](#) 1 (50) seule,
- G=20 : Allumage de la LED 2 (48) seule,
- B=30 : Allumage de la LED 3 (46) seule,
- X : éteindre les trois LEDs.



Le programme Matlab

```
close all; clc; clear all;

%% Ouverture de la Caméra (Multi-sources)
Type=1;      % 0(GRAY), 1(RGB)
Source=2;    % 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;
im=zeros(M,N,3);
DET_RGB0=zeros(1,3);
im_RGB=zeros(M,N,3);

%% Détection & Transfert vers Arduino
while 1
    %% 1. Lecture de l'image courante
    im0=getsnapshot(cam);
    im(:,:,1)=imresize(im0(:,:,1),[M N]);
    im(:,:,2)=imresize(im0(:,:,2),[M N]);
    im(:,:,3)=imresize(im0(:,:,3),[M N]);
    %% 2. Détection de l'objet (par couleur)
    Seuil=0.1; % Seuil de conversion au format binaire
    numPix=10; % Seuil de comptage des pixels allumés
    [im_RGB(:,:,1), DET_RGB0(1)]=getObj(im, Seuil, 'R');
    [im_RGB(:,:,2), DET_RGB0(2)]=getObj(im, Seuil, 'G');
    [im_RGB(:,:,3), DET_RGB0(3)]=getObj(im, Seuil, 'B');
    RGB8= DET_RGB0
    % Conversion de format
    DET_RGB=double(DET_RGB0>numPix)
    DET_RGB(1)=10*DET_RGB(1); % Composante R: Valeur 10
    DET_RGB(2)=20*DET_RGB(2); % Composante G: Valeur 20
    DET_RGB(3)=30*DET_RGB(3); % Composante B: Valeur 30
    %% 3. Transfert vers la carte Arduino
```



```
for j=1:3
    if DET_RGB(j)
        fprintf(SerialCOM,'%d\n',DET_RGB(j)); pause(0.1);
    end;
end;
%% 4. Affichage
figure(1);
subplot(121); imshow(im/255); title('Image Originale','fontsize',16);
subplot(122); imshow(im_RGB); title('Image Résultante','fontsize',16);
end
```

Le programme Arduino

```
#define LEDPin1 50
#define LEDPin2 48
#define LEDPin3 46

// Valeur acquise
int DataDet=0;
const int R=10;
const int G=20;
const int B=30;

void setup()
{
    // Init port série
    Serial.begin(9600);
    // Init des LEDs
    pinMode(LEDPin1, OUTPUT);
    pinMode(LEDPin2, OUTPUT);
    pinMode(LEDPin3, OUTPUT);
    // Initialisation
    digitalWrite(LEDPin1, LOW);
    digitalWrite(LEDPin2, LOW);
    digitalWrite(LEDPin3, LOW);
}

void loop()
```



```
{
  // Lecture du détecteur
  DataDet = Serial.parseInt();
  // Controle des LEDs
  switch (DataDet)
  {
    case R:
      digitalWrite(LEDpin1, HIGH);
      digitalWrite(LEDpin2, LOW);
      digitalWrite(LEDpin3, LOW);
      break;

    case G:
      digitalWrite(LEDpin1, LOW);
      digitalWrite(LEDpin2, HIGH);
      digitalWrite(LEDpin3, LOW);
      break;

    case B:
      digitalWrite(LEDpin1, LOW);
      digitalWrite(LEDpin2, LOW);
      digitalWrite(LEDpin3, HIGH);
      break;
    default:
      digitalWrite(LEDpin1, LOW);
      digitalWrite(LEDpin2, LOW);
      digitalWrite(LEDpin3, LOW);
      break;
  }
}
```

[Traitement d'Images | Matlab](#)

Click to rate this post!

[Total: 1 Average: 5]