

# Guide Capteur de Couleur

## Préparation du système :

1. Réaliser le câblage (voir image).
2. Lancer le logiciel Arduino.
3. Brancher le câble d'alimentation de l'Arduino au PC.
4. S'assurer que le logiciel Arduino est connecté au bon port (voir FAQ).
5. Ouvrir le moniteur série (voir FAQ).
6. Étalonner le capteur sur une surface blanche opaque.  
(Si vous n'avez pas eu le temps d'étalonner le capteur veuillez appuyer sur le bouton reset)
7. Le capteur est prêt à fonctionner.

## FAQ :

### Quelle logiciel lancer ?

arduino.exe



### Quelle câble d'alimentation brancher ?

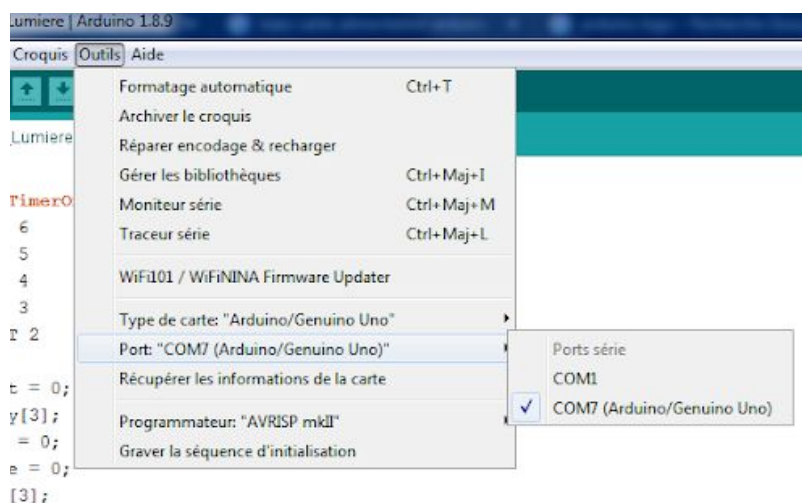
Le câble USB 2.0 A/B



### Quelle est le bon port pour l'Arduino ?

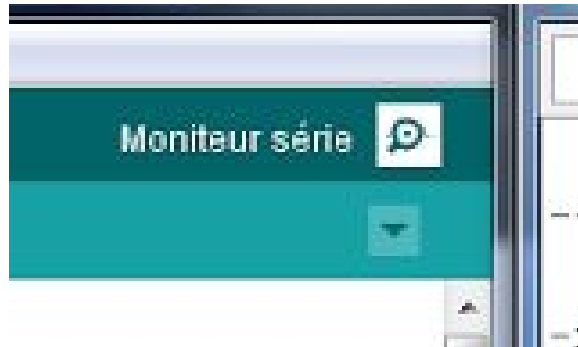
Il faut avoir cet affichage

(le numéro du port n'est pas important .)



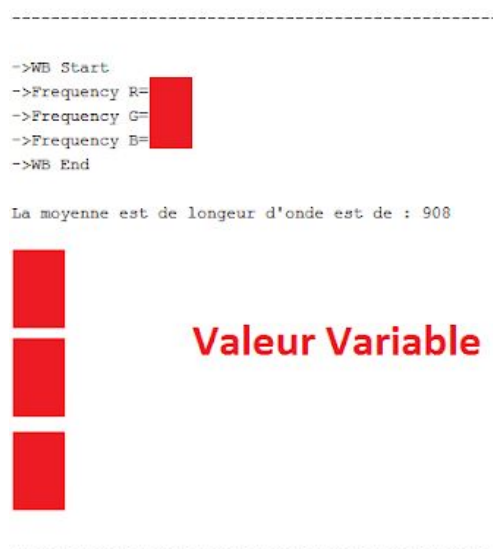
## Où trouver le moniteur série ?

Le bouton se trouve en haut à droite .



## Comment étalonner ?

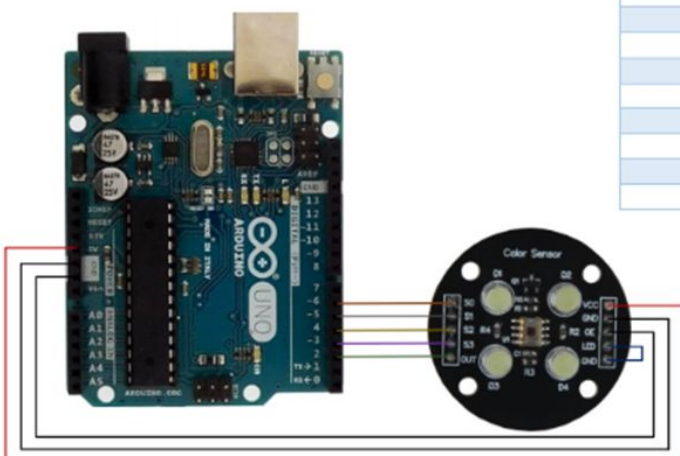
Il faut placer la surface blanche non réfléchissante au centre du capteur et attendre la fin de l'étalonnage .



## Image :

Table de correspondance :

Arduino	Embase PS2
6	S0
5	S1
4	S2
3	S3
2	OUT
+5V	VCC
GND	GND
GND	OE



Color	Color name	(R,G,B)
	Black	(0,0,0)
	White	(255,255,255)
	Red	(255,0,0)
	Lime	(0,255,0)
	Blue	(0,0,255)
	Yellow	(255,255,0)
	Cyan	(0,255,255)
	Magenta	(255,0,255)
	Silver	(192,192,192)
	Gray	(128,128,128)
	Maroon	(128,0,0)
	Olive	(128,128,0)
	Green	(0,128,0)
	Purple	(128,0,128)
	Teal	(0,128,128)
	Navy	(0,0,128)

## Code:

```
#include <TimerOne.h>
#define S0 6
#define S1 5
#define S2 4
#define S3 3
#define OUT 2

int g_count = 0;
int g_array[3];
int g_flag = 0;
int moyenne = 0;
float g_SF[3];

// Initialisation des pin
void TSC_Init()
{
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);
  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);
}

// Filtre les couleur
void TSC_FilterColor(int Level01, int Level02)
{
  if (Level01 != 0)
    Level01 = HIGH;
  if (Level02 != 0)
    Level02 = HIGH;
  digitalWrite(S2, Level01);
  digitalWrite(S3, Level02);
}

// Ajout +1 a g_count
void TSC_Count()
{
  g_count ++;
}

// Appel des dif couleur
void TSC_Callback()
{
  switch (g_flag)
  {
  case 0:
    Serial.println(" ");
    Serial.println("-----");
    Serial.println(" ");
    Serial.println("->WB Start");
    TSC_WB(Low, Low); // Filtre sans rouge
    break;
  }
```

```

case 1:
Serial.print("->Frequency R=");
Serial.println(g_count);
g_array[0] = g_count;
TSC_WB(HIGH, HIGH); // Filtre sans vert
break;

case 2:
Serial.print("->Frequency G=");
Serial.println(g_count);
g_array[1] = g_count;
TSC_WB(LOW, HIGH); // Filtre sans bleu
break;

case 3:
Serial.print("->Frequency B=");
Serial.println(g_count);
Serial.println("->WB End");
g_array[2] = g_count;
TSC_WB(HIGH, LOW); // Pas de filtre

Serial.println(" ");
moyenne = ((g_array[0] + g_array[1] + g_array[2]) / 3);
Serial.print("La moyenne est de longueur d'onde est de : ");
Serial.println(moyenne);
Serial.println(" ");
break;

default:g_count = 0;
break;
}
}

//Calcule du blanc dans l'environnement
void TSC_WB(int Level0, int Level1) // Balance des blancs
{
g_count = 0;
g_flag ++;
TSC_FilterColor(Level0, Level1);
Timer1.setPeriod(1000000);
}

//
void setup() {
TSC_Init();
Serial.begin(9600);
Timer1.initialize();
Timer1.attachInterrupt(TSC_Callback);
attachInterrupt(0, TSC_Count, RISING);
delay(4000);
for (int i = 0; i < 3; i++)
Serial.println(g_array[i]);
g_SF[0] = 255.0 / g_array[0]; // valeur R
g_SF[1] = 255.0 / g_array[1]; // valeur G
g_SF[2] = 255.0 / g_array[2]; // valeur B
Serial.println(" ");
Serial.println(g_SF[0]);
Serial.println(g_SF[1]);
Serial.println(g_SF[2]);
Serial.println(" ");
}

//
void loop()
{
g_flag = 0;
for(int i = 0; i < 3; i++)
Serial.println(int(g_array[i] * g_SF[i]));
delay(4000);
}

```