

Affichage d'une image d'introduction sur l'écran de l'EFIS ou de l'EMS

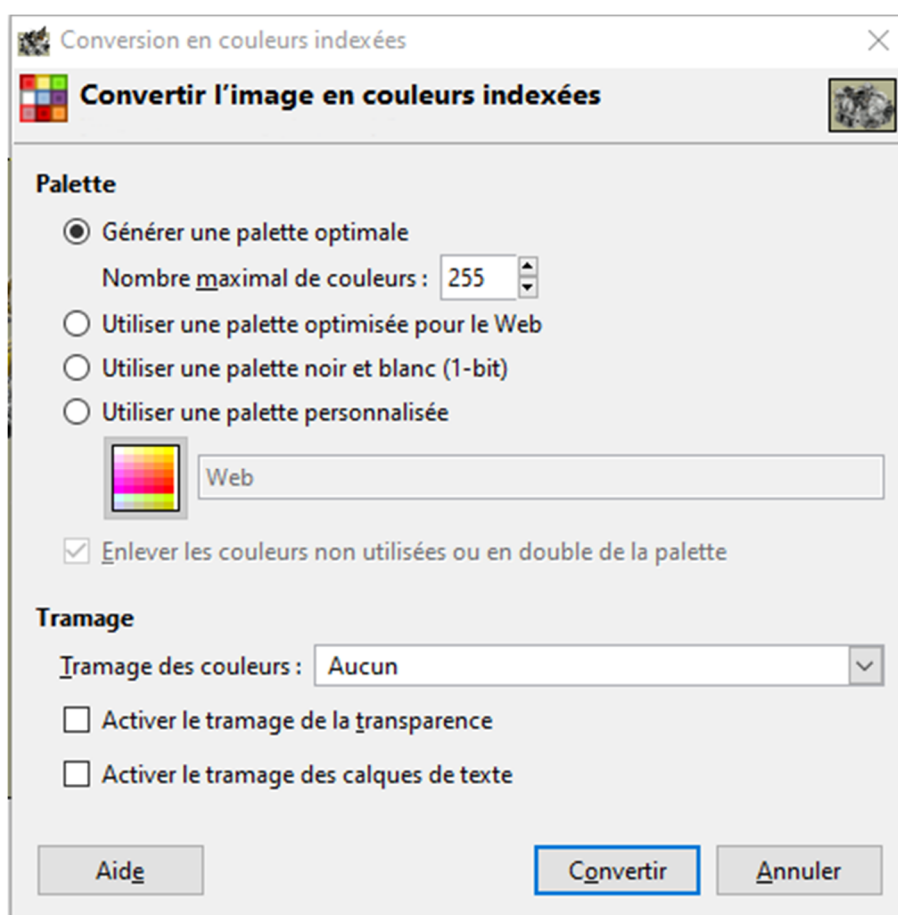
Introduction

Le traitement de l'image et sa conversion au format voulu (.xpm) font appel au logiciel GIMP qui doit donc préalablement être installé. GIMP est un outil d'édition et de retouche d'image gratuit et libre. Une fois l'image traitée et convertie au format .xpm, le fichier .xpm résultant (qui est au format texte) est copié-collé dans le fichier source (en C) d'une « moulinette » qui tourne sur une carte Teensy 4.1, et cette « moulinette » converti ce fichier xpm en un fichier binaire enregistré sur une carte microSD. Ce fichier binaire est ensuite directement affichable sur un écran au moyen de quelques lignes de code très simples.

Traitement et conversion

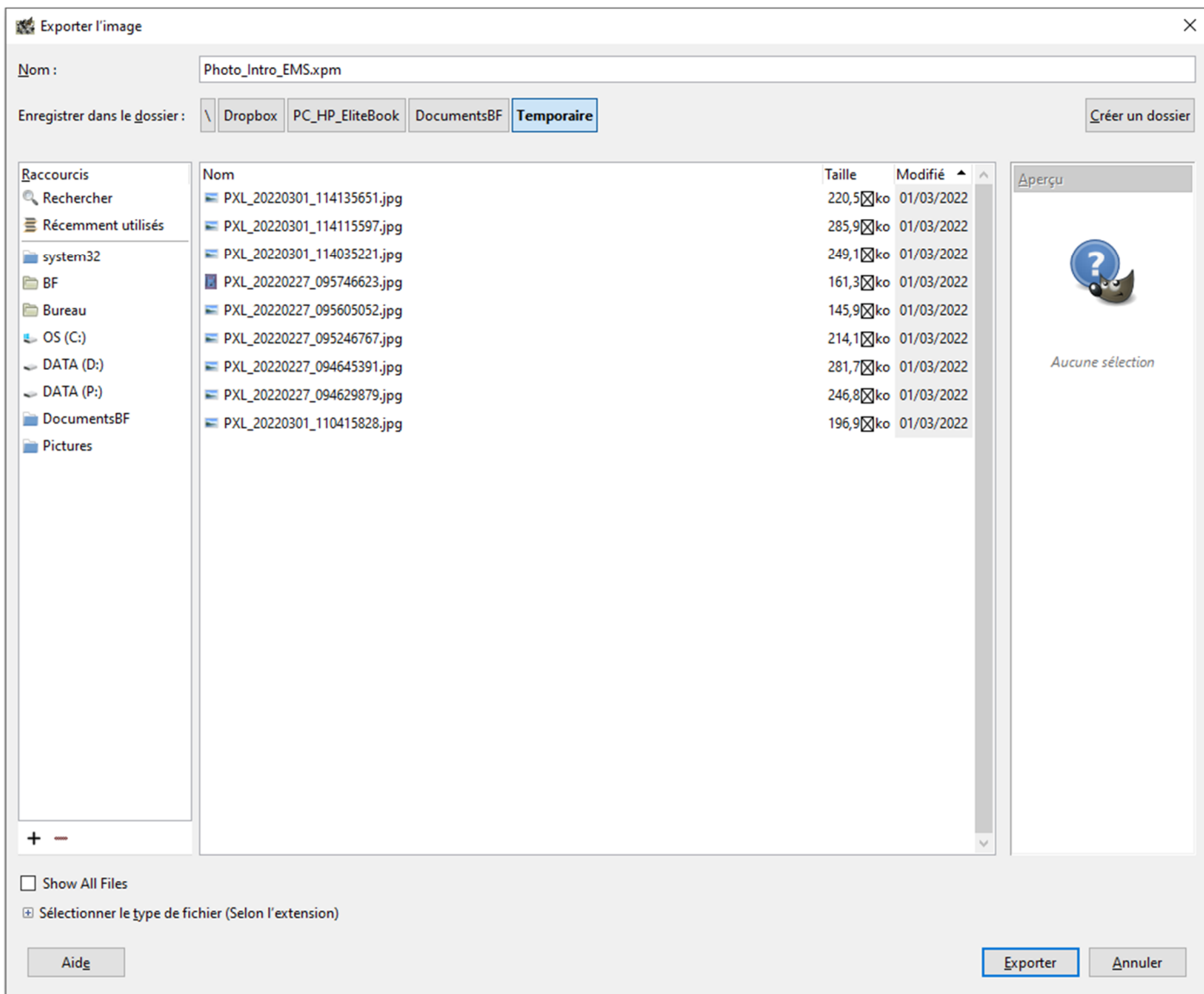
L'image choisie est chargée dans GIMP, puis retouchée à volonté.

Dans la commande « Image » → « Mode » → « Couleurs indexées... » cocher l'option « Générer une palette optimale ». Fixer le nombre maximal de couleurs à 255. Laisser le tramage sur Aucun, et ne rien cocher d'autre. Puis cliquer sur le bouton Convertir. Pourquoi 255 couleurs au maximum ? Parce que d'une part la mémoire graphique du RA8875 ne permet pas d'afficher simultanément plus de 255 couleurs sur chaque layer en mode 800 x 480. D'autre part cela permet un affichage raisonnablement rapide (ce contrôleur graphique n'est pas conçu pour afficher instantanément des images Bitmap). Enfin, avec 255 couleurs, sur un écran de petite taille, la qualité visuelle reste parfaite.



L'image ainsi convertie est maintenant codée sur 255 couleurs. Il ne reste plus qu'à l'exporter au format XPM.

Dans la commande « Fichier » → « Exporter sous... », choisir le dossier dans la case de gauche, puis saisir le nom souhaité dans la case « Nom » en haut de la boîte de dialogue, **sans oublier l'extension .xpm**, et cliquer sur le bouton « Exporter ». Dans la boîte de dialogue « Seuil alpha », laisser la valeur 127.



Le fichier .xpm obtenu doit maintenant être ouvert dans un éditeur de texte comme Notepad++ (puissant éditeur gratuit et libre), car une petite correction est parfois nécessaire.

Si le début du fichier se présente ainsi, aucune correction n'est nécessaire :

```

D:\Dropbox\PC_HP_EliteBook\DocumentsBF\912-UL_engine-pic-download.xpm - Notepad++
Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramètres  Outils  Macro  Exécution  Modules d'extension  Documents  ?
Rotax912_255couleurs.xpm  912-UL_engine-pic-download.xpm
1  /* XPM */
2  static char * D:\Dropbox\PC_HP_EliteBook\DocumentsBF\912_UL_engine_pic_download_xpm[] = {
3  "420 373 255 2",
4  "  c #000200",
5  ".  c #080200",
6  "+  c #030602",
7  "@  c #070A06",
8  "#  c #0E0806",
9  "$  c #0D100C",
10 "%  c #140F0E",
11 "&  c #1F0D01",
12 "*"  c #0D1114",

```

La ligne numéro 3 contient dans l'ordre la largeur de l'image en pixel (420), sa hauteur (373), le nombre total de couleurs de l'image (255), et le nombre de caractères ASCII requis pour donner un nom de code à chaque couleur, ici 2 caractères. En effet, seuls 92 des caractères ASCII standards et imprimables sont utilisés.


```

214 "[+ c #95B5B8",
215 ")+ c #ADAFAC",
216 "|+ c #B1AFA1",
217 "1+ c #B3B391",
218 "2+ c #BFB08A",
219 "3+ c #B0B2AE",
220 "4+ c #AFB2B5",
221 "5+ c #B7B39F",

```

Le coin supérieur gauche de notre image correspond effectivement à un fond uni gris beige clair :



On pourrait imaginer fournir directement ce fichier .xpm à notre logiciel d'EMS. Mais cela alourdirait beaucoup ce logiciel. En effet, il faudrait lui ajouter des routines un peu complexes de lecteur et de décodage des lignes de caractères xpm, et de conversion des couleurs 24 bits (RGB888) en couleurs 16 bit (RGB565), les seules que le RA8875 sait afficher. Donc on passe d'abord le fichier xpm « à la moulinette » pour le convertir en un fichier binaire. Il suffit ensuite d'ajouter quelques lignes très simples de code au logiciel (EMS ou autre) pour qu'il puisse afficher une photo à partir de ce fichier binaire.

La moulinette

Un exemple de code est téléchargeable [ici](#).

Toutes les lignes du fichier .xpm sont tout simplement copiées collées dans ce sketch

Selon le nom du fichier .xpm qui figure à la ligne 18, et le nom souhaité pour le fichier binaire, quelques modifications sont nécessaires.

Ce code doit être téléversé sur une Teensy 4.1, avec une carte Micro SD (formatée ExFAT) insérée dans le slot prévu à cet effet.

Après exécution, la carte Micro SD contient le fichier binaire. Dans cet exemple « ROTAX912.out »

L'affichage final

Pour afficher ce fichier, les quelques lignes de code à ajouter au programme sont les suivantes. On peut bien sûr raffiner le procédé en affichant un sablier sur une layer en premier plan, tandis que la photo s'affiche de façon cachée sur la layer d'arrière-plan (cet affichage pixel par pixel prend environ 2 secondes...), puis repasser la layer de la photo au premier plan, pour avoir une apparition instantanée de la photo. Avec ce code la photo s'affiche dans le coin supérieur gauche de l'écran, il est simple de modifier le code pour obtenir un affichage centré.

```
#include <SD.h> // Bibliothèque gérant la carte micro SD
```

```
File fichier; // Création d'un objet de type File, nommé « fichier »
```

```
#define largeurPhoto 410 // Taille en pixels de la photo à afficher en introduction
```

```
#define hauteurPhoto 320
void setup() {
SD.begin(BUILTIN_SDCARD); // Initialisation de la carte SD
photo(); // affiche la photo
}
void photo() { // Routine de lecture du fichier binaire et d'affichage des pixels composant la photo
fichier = SD.open("ROTAX912.out", FILE_READ);
if(fichier)
{
  unsigned char LSB, MSB;
  uint16_t couleur;
  for (uint16_t j=1; j<=hauteurPhoto; j++)
  {
    for (uint16_t i = 1; i<=largeurPhoto; i++)
    {
      LSB = fichier.read();
      MSB = fichier.read();
      couleur = LSB + (MSB<<8);
      tft.drawPixel(i,j,couleur);
    }
  }
  fichier.close();
}
else {tft.println("Erreur a l'ouverture du fichier ROTAX912.out"); delay(2000);}
}
```