# 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 »  $\rightarrow$  « Mode »  $\rightarrow$  « 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.

🗱 Conversion en couleurs indexées	$\times$
Convertir l'image en couleurs indexées	the second
Palette	
Of Générer une palette optimale	
Nombre <u>m</u> aximal de couleurs : 255	
O Utiliser une palette optimisée pour le Web	
<ul> <li>Utiliser une palette noir et blanc (1-bit)</li> </ul>	
<ul> <li>Utiliser une palette personnalisée</li> </ul>	
Web	
Enlever les couleurs non utilisées ou en double de la palette	
Tramage	
<u>Tramage des couleurs :</u> Aucun	$\sim$
Activer le tramage de la <u>t</u> ransparence	
Activer le tramage des calques de texte	
Aid <u>e</u> C <u>o</u> nvertir <u>A</u> nn	uler

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, <u>sans oublier l'extension .xpm</u>, et cliquer sur le bouton « Exporter ». Dans la boîte de dialogue « Seuil alpha », laisser la valeur 127.

🗱 Exporter l'image X									
<u>N</u> om :	Photo_Intro_EMS.xpm								
Enregistrer dans le <u>d</u> ossier :	\ Dropbox	PC_HP_EliteBook	DocumentsBF	Temporaire					<u>C</u> réer un dossier
Raccourcis Rechercher Récemment utilisés system32 BF Bureau OS (C:) DATA (D:) DATA (P:) DocumentsBF Pictures	Nom PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022 PXL_2022	0301_114135651.jpg 0301_114115597.jpg 0301_114035221.jpg 0227_095746623.jpg 0227_095246767.jpg 0227_094645391.jpg 0227_094629879.jpg 0301_110415828.jpg				Taille         220,5∑ko         285,9∑ko         249,1∑ko         161,3∑ko         214,1∑ko         246,8∑ko         196,9∑ko	Modifié	A	vacune sélection
<ul> <li>Show All Files</li> <li>Sélectionner le type de fi</li> </ul>	chier (Selon l'e	xtension)							
Aid <u>e</u>							I	<u>E</u> xporter	Annuler

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 :



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.

Par exemple sur la ligne 8 de la capture d'écran ci-dessus, on voit que le nom de code «# » (le caractère '#' suivi du caractère espace) désigne la couleur RGB 24 bits 0x0E0806.

Parfois, il se peut que le fichier .xpm commence ainsi :

D:\Dropbox\PC\_HP\_EliteBook\DocumentsBF\Rotax912\_255couleurs.xpm - Notepad++

<b>Fichier</b>	<u>É</u> dition <u>R</u> echerche <u>Affichage</u> Encodage <u>Langage</u> Paramètres <u>O</u> utils <u>M</u> acro Exécution Modules d'extension <u>D</u> o
🕞 占 (	🗄 🛍 📴 🕼 🚔   🕹 🛍   Ə 🥑   🏙 🍖   🍕 👒   🍱 🖼   🎫 11 🎼 💷 💹 🚳 🖉   🗉 🔍 🔍
📄 Rotax9	112_255couleurs.xpm 🔀 🔚 912-UL_engine-pic-download.xpm 🔀
1	/* XPM */
2	<pre>static char * D:\Dropbox\PC_HP_EliteBook\DocumentsBF\Rotax912_255couleurs_xpm[] = {</pre>
3	"410 320 255 2",
4	" c None",
5	". c #000200",
6	"+ c #080200",
7	"@ c #030602",
8	"# c #070A07",
9	"\$ c #0E0806",
10	"% c #0B0B00",

Sur la ligne 4, on lit : " c None", ce qui signifie que l'image n'utilise en fait que 254 couleur, le nom de code « » (double espace) n'est attribué à aucune couleurs. Une correction s'impose : le nombre de couleurs doit être modifié à 254, et la ligne 4 doit être supprimée. Ce qui donne :

\*D:\Dropbox\PC\_HP\_EliteBook\DocumentsBF\Rotax912\_255couleurs.xpm - Notepad++

<u>Fichier</u>	<u>É</u> ditio	n	<u>R</u> echerche	<u>A</u> ffichage	E <u>n</u> codage	<u>L</u> angage	<u>P</u> aramètres	<u>O</u> utils	<u>M</u> acro	E <u>x</u> écution	Modules d'extension	D
🕞 占 🛛	-		è 🕞 🎒	* h (	) <b>) c</b>	# b	🤏 🦂   🖪	<b>-</b>	ə 11 耳	E 🐖 💹 🗓	š 🔊 🖿 💿 🗐	
📄 Rotax9	12_25	5соц	uleurs.xpm 🔀	912-UL 🔚	_engine-pic-do	wnload.xpm	×					
1	/* 1	XPM	1 */									
2	<pre>2 static char * D:\Dropbox\PC HP EliteBook\DocumentsBF\Rotax912 255couleurs xpm[] = {</pre>											
3	3 "410 320 254 2",											
4												
5	۳.	С	#000200	",								
6	"+	С	#080200	",								
7	"@	C	#030602	",								
8	"#	С	#070A07	",								
9	"Ş	С	#0E0806	",								

Le fichier .xpm ainsi corrigé comporte donc 254 lignes de codage ASCII des couleurs utilisées. Puis suivent 320 lignes de chacune 410 couples de caractères ASCII, soit 820 caractères. Chaque couple code la couleur d'un pixel.

Sur la capture d'écran ci-dessous, on voit la fin des 254 lignes de codage ASCII, et le début des lignes de « pixels ».

253	"A+ c #DEEODD",
254	"B+ c #EDE4C1",
255	"C+ c #E3E5E2",
256	"D+ c #E9EBE8",
257	"E+ c #F2F4F1",
258	"F+ c #F9FBF8",
259	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
260	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
261	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
262	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
263	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
264	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+
265	"1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+

Le début de ces lignes correspond au coin supérieur gauche de l'image, la couleur codée par le couple ASCII « 1+ » correspond à la couleur RGB 24 bits 0xB3B391 (un gris beige clair), voir capture ci-dessous, ligne 217 :

214	"[+	с	#95B5B8",
215	"}+	С	#ADAFAC",
216	" +	С	<pre>#Blafal",</pre>
217	"1+	С	#B3B391",
218	"2+	С	#BFB08A",
219	"3+	С	#BOB2AE",
220	"4+	С	#AFB2B5",
221	"5+	С	#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 <u>ici</u>.

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++)</pre>
   {
    for (uint16_t i = 1; i<=largeurPhoto; i++)</pre>
      {
       LSB = fichier.read();
       MSB = fichier.read();
       couleur = LSB + (MSB<<8);</pre>
       tft.drawPixel(i,j,couleur);
      }
   }
  fichier.close();
 }
else {tft.println("Erreur a l'ouverture du fichier ROTAX912.out"); delay(2000);}
}
```