

PROTOCOLE DE TRANSMISSION D'IMAGES NUMÉRIQUES

F6CTE, Patrick LINDECKER

Dans cet article, je décris un protocole de transmission d'images numériques appelé "Run" qui a pour principales originalités, d'une part, de pouvoir s'intégrer dans un texte et, d'autre part, de pouvoir être décodé à n'importe quel moment en cours de la transmission. Les modes de transmission supportant ce protocole sont, actuellement, le Packet (300 et 1200 bauds), le PSK63F et le PSK220F.

Le logiciel permettant cette fonctionnalité s'appelle "Multipsk" : c'est un programme téléchargeable sur le site Internet de l'auteur :

<http://f6cte.free.fr>

A la fin de ce papier sont présentées quelques copies d'écran du programme.

SSTV numérique : objectifs généraux du protocole de compression et de transmission d'image "Run" - Version 1
(Copyright © 2005 Patrick Lindecker Bures-sur-Yvette France)

1) Le but n'est pas ici de transmettre le fichier binaire d'une image déjà compressé mais

- de charger un fichier image sous la forme d'un tableau de "pixels",

- de le compresser de la manière la plus appropriée pour des transmissions radio,

- d'assurer sa transmission, puis sa réception.

2) La transmission de l'image compressée doit se faire, préférentiellement, par un mode numérique ne produisant aucune erreur de transmission. Le Packet en mode connecté répond à ce besoin. La transmission se fera donc entre 2 OM en mode Packet "connecté".

Cependant, ce protocole pourra être également utilisé via des modes de transmission fiables (comme le PSK63F ou le PSK220F) qui permettront tout de même des transmissions d'images (même si, du fait des erreurs, l'image peut présenter des imperfections).

3) Un SWL ou un OM à l'écoute doit pouvoir décodé l'image

numérique, mais avec un risque d'erreurs puisque les données mal reçues ne seront pas répétées (en Packet), donc il faut nécessairement que

- les changements de lignes puissent être identifiés et définissent le type d'image compressé (couleurs, nuances de gris ou noir et blanc),

- les numéros de lignes soient transmis.

L'unité de compression choisie ici est la ligne, ce qui exclut un codage de type JPEG (transformation DCT sur des blocs de 8x8 pixels).

4) Un SWL ou un OM à l'écoute doit pouvoir décodé l'image alors même que la transmission a déjà démarré, ce qui exclut des codages prenant en compte des données précédemment transmises (codage prédictif), hormis celles de la ligne courante.

5) La transmission pourra se faire en couleurs, en nuances de gris ou en noir et blanc (du type "fax HF").

6) En couleurs et en nuances de gris, la compression ne doit affecter que très faiblement la qualité de l'image (différence entre l'original et la copie non discernable sans examen précis). En noir et blanc, il ne doit pas y avoir de perte due à la compression (copie identique à l'original, ceci pour transmettre, par exemple, des schémas électroniques).

7) La taille de l'image transmise doit être comprise entre 8x6 et 320x256 (ce dernier format étant celui utilisé en SSTV analogique classique).

Nota 1 : le taux de compression est toujours rapporté à une

image où chaque pixel est codé sur 24 bits (format classique BMP "bitmap 24 bits").

Par exemple, une image couleur 40 x 30 sera codée initialement (avant compression) sur 28800 (40 x 30 x 24) bits.

Nota 2 : ces objectifs limitent le choix de la méthode de compression à des méthodes simples. Celle choisie est appelée RLE ("Run Length Encoding" ou, en français "Codage par plage") d'où le nom de ce protocole : "Run".

Les taux de compression obtenus sont donc intéressants mais modestes : typiquement 3,5 pour des images couleurs, 7 en nuances de gris, 25 en noir et blanc, plus faible pour des images photographiques et plus fort pour des images synthétiques. Par ailleurs, plus l'image est petite, plus le taux de compression est faible, et inversement.

Spécifications détaillées du protocole de compression et de transmission d'image "Run"
1) Procédé pour la compression et la transmission d'une image en couleurs

Notations

<L> : largeur de l'image (limitée à 320 pixels) exprimée par 3 chiffres

<H> : hauteur de l'image (limitée à 256 pixels) exprimée par 3 chiffres

Y : composante "Luminance Y" sur 8 bits initialement

Cb : composante "Chrominance Cb" sur 8 bits initialement

Cr : composante "Chrominance Cr" sur 8 bits initialement

Opérations :

1) Après avoir ouvert le fichier au format BMP (ou JPG mais transformé par le programme en "bitmap"), on commence par transformer les pixels exprimés à l'aide des 3 couleurs fondamentales R (Rouge), V (Vert), B (Bleu) en composantes Luminance Y et Chrominances Cb, Cr (comme en compression JPEG).

2) Pour chacune des composantes Y, Cb, Cr, on passe d'une amplitude sur 8 bits à une amplitude sur 5 bits (en deçà de 5 bits l'image est altérée), soit pour 0 : "00000" et pour 31 : "11111".

Nota : la faible dégradation de l'image tient donc au passage de 16.777.216 couleurs à 32.768 couleurs, en mode couleurs ou de 256 à 32 nuances de gris en mode "dégradés de gris".

3) L'unité de base supportant la compression étant la ligne, on répète cette opération autant de fois qu'il y a de pixels sur la ligne.

A ce niveau, on dispose donc pour chaque ligne de 3 chaînes binaires (une par composante Y, Cb et Cr) composées d'amplitudes comprises entre 0 et 31. Pour une composante (Y,Cb ou Cr), il y a autant d'amplitudes que de pixels dans la ligne.

Exemple : "1 1 1 1 1 1 5 5 6 4 2 2 2 2...".

4) Pour chacune des 3 composantes de chaque ligne, on effectue le traitement RLE ("Run Length Encoding") suivant :

On détermine le nombre d'amplitudes (N) successives de même valeur et on fait précéder cette valeur du nombre N et du bit 0. De même, on détermine le nombre (N) d'amplitudes successives correspondant à des valeurs toujours différentes et on fait précéder ces valeurs du nombre N et du bit 1.

Le nombre N maximum correspond à $2^L - 1$ (par exemple, "111" si $L=3$). Le nombre $N=0$ n'est jamais, évidemment, utilisé. La longueur L du nombre N peut varier entre 3 et 6 bits. Chaque longueur L est testée vis-à-vis du critère de meilleur taux de compression puis stockée sur 2 bits ("00" pour 3 bits à "11" pour 6 bits).

Exemple : "1 1 1 1 1 1 5 5 6 4 2 2 2..." s'écrit (en décimal) après traitement : "0 7 1 0 2 5 1 2 6 4 0 4 2, "0 7 1" s'écrivant en binaire, si $L=3$: "00000001" et si $L=4$: "0011100001".

5) Une ligne, sous forme binaire (appelée "LB" par la suite), est composée des éléments suivants :

* le signal de début d'une ligne en couleurs :

"10000000000000000001" (19 "0"),

* le numéro de ligne (entre 1 et 256 sur 8 bits : "00000000" pour 1 et "11111111" pour 256),

* le nombre N (2 bits),

* la chaîne binaire luminance Y,

* la chaîne binaire chrominance Cb,

* la chaîne binaire chrominance Cr.

6) Il faut maintenant transmettre l'image sous forme binaire.

Avant cela, il faut prévenir le correspondant ou ceux à l'écoute de l'envoi d'une image. Pour cela, l'image doit être précédée d'un préfixe, du type, "Run<CHR(1)><L>x<H>C" pour une image couleurs ("C") de dimensions <L>x<H>, utilisant le protocole "Run".

Par exemple, le préfixe "Run320x256C" pourrait être généré par le programme.

Sur réception de ce préfixe, le programme passe en mode "décodage d'une image numérique" et n'interprète plus les bits reçus comme étant du texte.

Note : ce préfixe a uniquement un caractère informatif pour les OM et SWL à l'écoute. En effet, comme, par hypothèse, le décodage peut démarrer durant la transmission de l'image, d'autres éléments doivent permettre de retrouver les caractéristiques de l'image :

* les signaux de début de ligne

fournissent le type d'image (couleurs, nuances de gris ou noir et blanc) et permettent au programme de passer en mode "décodage d'une image numérique",

* le nombre de lignes est déduit de la réception du signal de fin d'envoi de l'image,

* la largeur des lignes en pixels est déduite au décodage.

Les règles sur ce préfixe sont les suivantes :

* les 3 premiers caractères imprimables ("Run") sont fixes. Ils sont suivis du caractère non imprimable fixe CHR(1). Ces 4 caractères permettent de discriminer l'envoi d'une image de celui d'un texte.

Nota : pour donner plus de clarté au texte et pour améliorer la synchronisation, on fera précéder ces caractères de six espaces.

* ils sont suivis du format de l'image qui est composé d'un nombre formé de 3 chiffres ("040", par exemple), représentant la largeur de l'image (<L>), suivi du signe "x" (petit "X") suivi d'un nombre formé de 3 chiffres ("030", par exemple), représentant la hauteur de l'image (<H>).

La largeur de l'image doit être comprise entre "008" et "320", sa hauteur entre "006" et "256",

* la lettre suivante (obligatoire) vaut "C" s'il s'agit d'une image couleurs, "G" pour une image en nuances de gris, "B" pour une image en noir et blanc.

Un espace est transmis après cette dernière lettre. Il est juste destiné au décodage de ce caractère.

Nota : en préalable au préfixe, on pourra, éventuellement, envoyer une description succincte de l'image ou un commentaire.

7) Une fois le préfixe envoyé, on envoie les lignes binaires (LB) de la ligne 1 à la dernière ligne (≤ 256) puis deux fois le signal de fin d'envoi de l'image avec le bit "0" entre les 2 :

"1000000000000000000000000001" (25 "0") puis "0" puis "1000000000000000000000000001".

Nota : au cours de la réception d'une image, si l'on n'a pas décodé de signaux de fin d'envoi ou de début de ligne durant

30 secondes, le programme repasse en réception de texte. Le décodage se fera, bien sûr, dans le sens inverse du codage, à partir de la chaîne binaire comprise entre deux signaux de début de ligne (cf. 5)) ou entre le dernier signal de début de ligne et le signal de fin d'envoi de l'image (cf. 7)).

II) Procédé pour la compression et la transmission d'une image en niveaux de gris
Préliminaire : on part toujours d'une image couleur 24 bits.

Il s'agit du même procédé que pour la compression d'images en couleurs, excepté les éléments suivants :

* seule la chaîne Luminance (Y) est transmise. Lors du décodage, on considérera que les chrominances (Cb et Cr) sont égales à 128,

* le signal de début d'une ligne en nuances de gris est :

"10000000000000000001" (18 "0"),

* le préfixe sera du type "Run<CHR(1)><L>x<H>G" ("Run320x256G", par exemple) (avec six espaces devant).

III) Procédé pour la compression d'une image et la transmission d'une image en noir et blanc (du type "Fax HF")
Préliminaire : on part toujours d'une image couleur 24 bits.

Opérations :

1) Après avoir ouvert le fichier au format BMP (ou JPG mais transformé par le programme en "bitmap"), on commence par transformer les pixels exprimés à l'aide des 3 couleurs fondamentales R (Rouge), V (Vert), B (Bleu) en composantes Luminance Y et Chrominances Cb, Cr (comme en compression JPEG).

2) Sur la composante Y, on définit un niveau binaire "0" si le niveau de Y est ≤ 127 et un niveau binaire "1" si le niveau de Y est ≥ 128 .

L'unité de base supportant la compression étant la ligne, on répète cette opération autant de fois qu'il y a de pixels sur la ligne. On va donc obtenir une suite binaire, par exemple :

"11111100000...".

3) On effectue le traitement RLE ("Run Length Encoding") suivant :

On détermine le nombre de bits (N) successifs de même valeur et on fait précéder cette valeur du nombre N et du bit 0. De même, on détermine le nombre de bits (N) successifs correspondant à des valeurs de bits toujours différentes et on fait précéder ces bits (en fait le premier bit suffit) du nombre N et du bit 1.

La longueur L du nombre N peut varier entre 3 et 6 bits. Chaque longueur L est testée vis-à-vis du critère de meilleur taux de compression puis stockée sur 2 bits ("00" pour 3 bits à "11" pour 6 bits).

S'il s'agit d'une chaîne de bits différents, on code le premier bit (les autres bits se déduisant trivialement).

Le nombre N maximum correspond à $2^L - 1$ (par exemple, "111" si $L=3$). Le nombre $N=0$ n'est jamais, évidemment, utilisé.

Si la longueur d'une chaîne de bits identiques est inférieure au maximum ($2^L - 1$), le bit suivant une chaîne de bits identiques n'est pas codé (puisque sa valeur est triviale) et de même pour une chaîne de bits différents.

Exemple :

"1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 0" s'écrit (en décimal) après traitement :

"0 7 1 0 4 1 1 2 1 1 2 1",

et en binaire, si $L=4$:

"001111 001001 100101 100101"

Nota : il y a, de ce fait, une ambiguïté sur la position du dernier bit. Un pixel noir supplémentaire peut donc, éventuellement, apparaître en fin de ligne.

Cependant, le programme pourra, après considération de plusieurs lignes, déterminer la vraie longueur de ligne et supprimer ce pixel.

4) Une ligne, sous forme binaire (appelée "LB" par la suite), est composée des éléments suivants :

* le signal de début d'une ligne en noir et blanc est :

"10000000000000000001" (17 "0"),

* le numéro de ligne (entre 1 et 256 sur 8 bits : "00000000" pour 1 et "11111111" pour 256),

* le nombre N (2 bits),

* la chaîne binaire luminance Y.

technique

5) Il faut maintenant transmettre l'image sous forme binaire. Avant cela, il faut prévenir le correspondant de l'envoi d'une image.

Pour cela, l'image doit être précédée d'un préfixe, par exemple : "Run320x256B" pour une image noir et blanc ("B") de dimensions 320x256, utilisant le protocole de compression "Run" (avec six espaces devant). Les règles sur ce préfixe ont été données précédemment.

6) Une fois le préfixe envoyé, on envoie les lignes binaires (LB) de la ligne 1 à la dernière ligne (≤ 256) puis deux fois le signal de fin d'envoi de l'image avec le bit "0" entre les 2 : "10000000000000000000000000000001" (25 "0") puis "0" puis "10000000000000000000000000000001".

Nota : au cours de la réception d'une image, si l'on n'a pas décodé de signaux de fin d'envoi ou de début de ligne durant 30 secondes, on repasse en réception de texte.

Lors du décodage, on considérera que les chrominances (Cb et Cr) sont égales à 128.

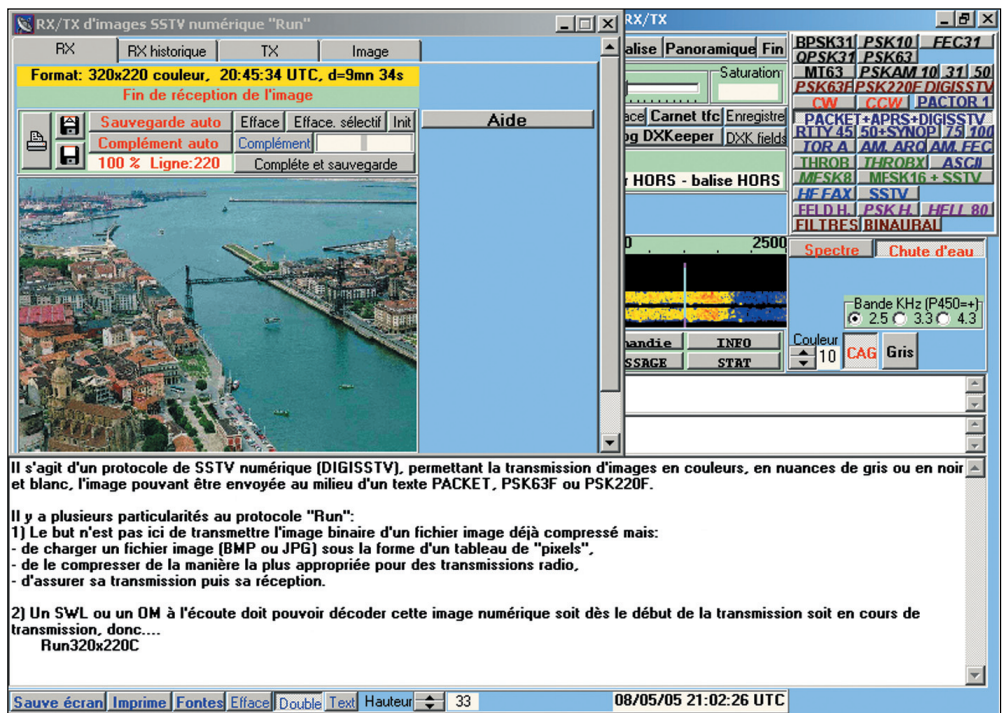
IV) Notes générales

1) Les 3 signaux de début de ligne et le signal de fin d'envoi de l'image (bit 1 X bit 1 avec X composé de 17, 18, 19 ou 25 bits "0") sont complètement liés au protocole décrit précédemment puisqu'ils permettent de détecter les différents types d'image (couleurs, nuances de gris et noir et blanc) ainsi que la fin d'envoi de l'image.

Donc, ces 4 séquences de bits doivent être uniques sur les modes de transmission (comme en PSK63F et en PSK220F) où le protocole "Run" est utilisé.

Nota : la probabilité de rencontrer ces séquences de bits est très faible et celle d'être validé par le "Chek-Sum" en Packet est totalement négligeable.

2) Actuellement la version Multipsk 3.9 implante ce protocole uniquement sur les modes PSK63F, PSK220F et Packet (300 et 1200 bauds).

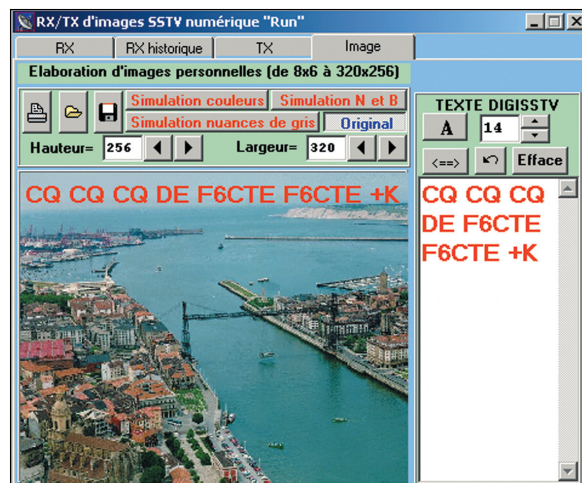
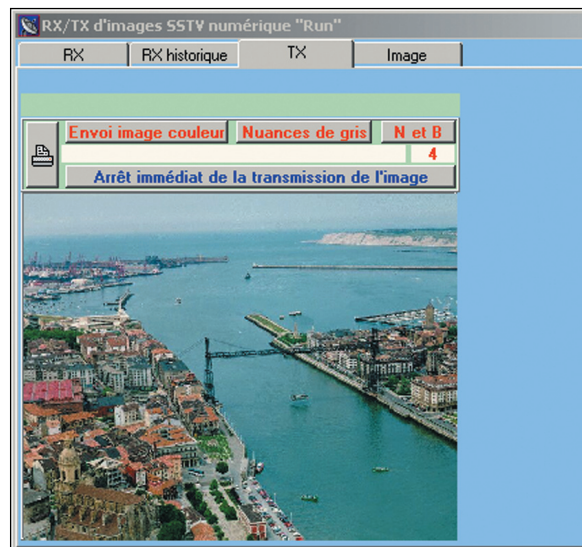


Il s'agit d'un protocole de SSTV numérique (DIGISSTV), permettant la transmission d'images en couleurs, en nuances de gris ou en noir et blanc, l'image pouvant être envoyée au milieu d'un texte PACKET, PSK63F ou PSK220F.

Il y a plusieurs particularités au protocole "Run":

- 1) Le but n'est pas ici de transmettre l'image binaire d'un fichier image déjà compressé mais:
 - de charger un fichier image (BMP ou JPG) sous la forme d'un tableau de "pixels",
 - de le compresser de la manière la plus appropriée pour des transmissions radio,
 - d'assurer sa transmission puis sa réception.

- 2) Un SWL ou un OM à l'écoute doit pouvoir décoder cette image numérique soit dès le début de la transmission soit en cours de transmission, donc....
Run320x220C



BRÈVES BXC

microphone oreillette

Le Coréen Samsung a fabriqué une oreillette Bluetooth pour téléphone mobile intégrant un microphone et un haut-parleur. Originalité: le microphone ne capte pas les sons véhiculés dans l'air, mais par la boîte crânienne du porteur. Cette technique dite de "conduction osseuse" permet de filtrer jusqu'à 90% des perturbations sonores pour ne transmettre que la parole du locuteur et ce dans les environnements les plus bruyants.

conduits de lumière

Des sociétés ont construit des conduits de lumière souples permettant de guider la lumière issue de la puce lumineuse placée sur son circuit imprimé jusqu'à un panneau pouvant être distant de 100 à 250 mètres.
Fabricant Lumex ou Optosource.

diode Schottky + transistor Mos

Toshiba a fabriqué un transistor Mos intégrant sur la même puce une diode Schottky, créant ainsi un nouveau composant baptisé MOSBD. Cette première mondiale permettrait d'augmenter le niveau d'intégration des applications portables et le rendement des convertisseurs de tension.