

**Patrick Lindecker (F6CTE) le 08 avril 2008**  
**Mis à jour le 22 août 2018**

# **LES IDENTIFICATEURS DE DIGIMODE**

## **Introduction**

L'identification d'un mode numérique amateur (appelé couramment « digimode ») ne peut pas se faire automatiquement par logiciel, ceci du fait de la diversité des modulations utilisées (BPSK, QPSK, FSK, MFSK, IFK, OOK...), des vitesses utilisées (de 1 à 9600 bauds), des codages utilisés (convolutionnels, Reed Solomon ou autre). D'autre part, il peut y avoir plusieurs transmissions de différents types sur la même bande. Cependant, dans le cas particulier d'une unique modulation et une seule transmission dans la bande de réception, un mode peut, éventuellement, être identifié automatiquement, comme par exemple le FSK sur RadioRaft de François Guillet, F6FLT.

Le problème de la reconnaissance automatique des digimodes, bien que passionnant, n'est pas soluble pour l'instant (peut-être le sera-t-il dans le futur avec des traitements basés sur des algorithmes d'intelligence artificielle). Il existe, cependant, des logiciels, comme "Signals Analyzer" qui aident à la détermination du mode utilisé.

Certains modes sont simples à identifier par l'utilisateur, soit parce que leur fréquence est bien connue (BPSK31 sur 14070 KHz, par exemple) soit parce que leur signature visuelle sur la « chute d'eau » ou acoustique est caractéristique (RTTY 45 bauds, par exemple).

Mais pour la plupart des autres digimodes (plus ou moins exotiques), il est très difficile de reconnaître le mode ou le sous-mode utilisé avec de simples traces acoustiques et/ou visuelles. D'où le besoin d'identificateurs de digimodes.

Actuellement, le seul identificateur officiel est celui qui définit le sous-mode SSTV analogique utilisé (Robot 36, Martin 1...). Il fonctionne parfaitement mais est limité à la SSTV.

En général, il est possible d'envoyer (à 20 mpm sur Multipsk) un petit texte CW avant chaque transmission du digimode. Mais ce procédé est rarement utilisé. De plus, la CW pas toujours comprise et, même si elle est comprise, le temps de latence avant le début de décodage cérébral fait perdre le début du message.

Ici sont donc présentés deux identificateurs de modes : le « RS ID » et le « Video ID » qui simplifient l'identification des transmissions en digimode, lorsqu'ils sont utilisés.

# Identificateur Reed-Solomon (« RS ID ») de mode et de fréquence

## Origine

L'identificateur "RS ID " (pour "Reed-Solomon Identifier") a été créé par l'auteur (Patrick Lindecker) et ajouté à la version 4.1.1 de Multipsk (en novembre 2006).

## Principale utilisation

L'identificateur "RS ID" permet d'identifier, automatiquement, toute transmission numérique faite dans un des modes RX/TX gérés par un logiciel de codage/décodage.

Nota: tous les RS ID sont détectés par Multipsk mais tous les modes ne sont pas décodés par Multipsk comme, par exemple, le FDMDV de Cesco (HB9TLK) ou les modes PSK\*\*RX\* de Fldigi. Chaque programme de décodage décode un sous-ensemble de l'ensemble des modes détectés à travers le RS ID.

Il est utilisé par les OMs pour les modes numériques dits « exotiques » (MFSK, Olivia, Contestia, THROB et beaucoup d'autres), c'est-à-dire presque tous ceux qui ne sont ni du RTTY ni du PSK31 ni du JT65 ni du FT8 ni de la SSTV... Cet identifiant est aussi utilisé dans les transmissions faites par les radios « OM » :

- <http://www.slowskanradio.com/>
- <http://swradiogram.net/>

## Fonctionnement

Sur réception d'un RS ID, il est détecté, d'un part, le mode utilisé et, d'autre part, la fréquence centrale de l'identificateur, qui est aussi la fréquence centrale de la transmission, avec une précision de +/- 2,7 Hz.

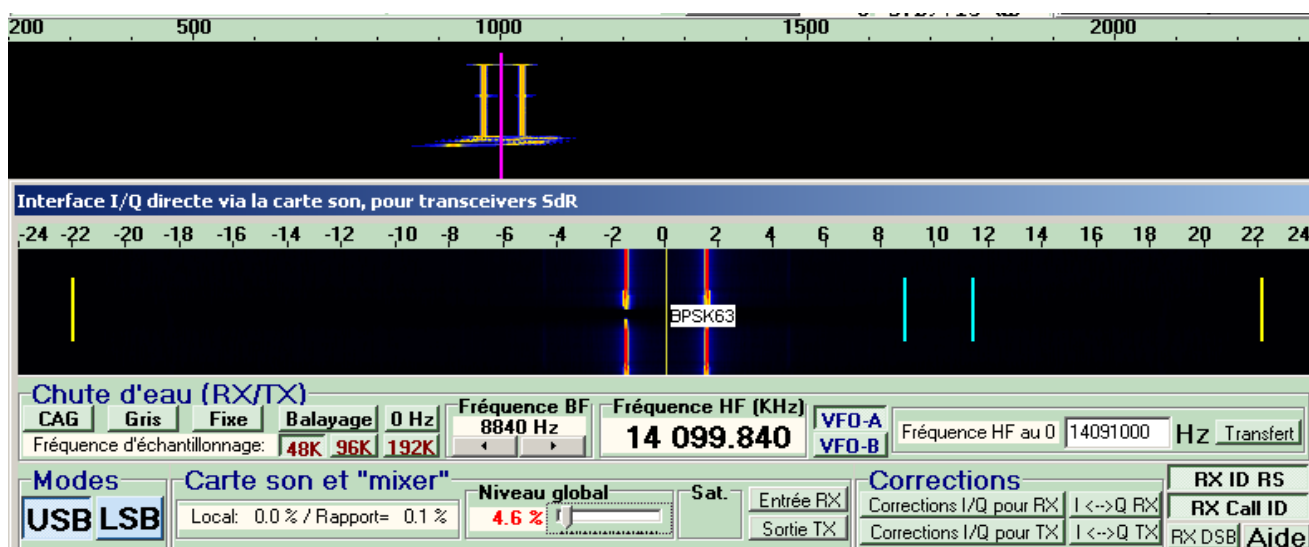
A la réception de cet identificateur, le logiciel de décodage commute automatiquement sur le mode et la fréquence reçus et décode immédiatement le QSO en cours ou l'appel général. Cet identificateur est transmis en 1,4 sec et occupe une largeur de bande de 172 Hz. Sa détection se fait jusqu'à un rapport Signal sur Bruit d'environ -16 dB, soit donc avec une sensibilité égale ou meilleure que celle de la plupart des modes numériques (RTTY, PSK31...), sauf certains comme les modes PSK10, THROB, THROBX ou JT65.

Nota: en conséquence, il pourrait, éventuellement, arriver que le RS ID soit détecté mais que l'appel ou le QSO ne soit pas décodé du fait d'un signal trop faible.

Cet identificateur peut être lancé, en priorité, avant chaque appel général ou en préalable à chaque réponse dans un QSO.

La recherche est, en général, faite sur la bande 200-2500 Hz. Pour cette largeur de bande, la charge CPU équivalente est négligeable. Mais la recherche peut également être faite sur une largeur de 44 KHz à partir d'un récepteur SdR. La charge CPU équivalente est alors d'environ 500 MHz. Voir ci-dessous une copie d'écran de l'interface SdR Multipsk, sur réception d'un RS ID.

Nota : les fausses détections sont théoriquement possibles, mais statistiquement extrêmement rares, ceci du fait d'une très forte fonction d'autocorrélation.

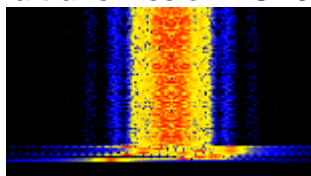


Pour l'instant, cinq programmes proposent le RS ID: PocketDigi, FDMDV, DM780, FLdigi et Multipsk.

Les programmes sources RS ID sont disponibles auprès de l'auteur (en Delphi et incluant la détection SdR).

La liste des identifiants RS ID (sous forme de numéros) associés aux modes numériques est donnée en Annexe 1.

Par exemple, voici le RS ID PSK63 (en forme de racine) reçu juste avant la transmission PSK63 elle-même (en forme de tronc).



## Principe

Chaque mode correspond à un numéro qui est transformé en une séquence Reed-Solomon particulière. Ce codage RS (  $RS(k=4,t=6)$  ) est défini par les paramètres  $k=4$  (nombre de bits par symbole),  $n=15$  ( $=2^k-1$ , nombre de symboles par séquence RS) et  $t=6$  (nombre maximum d'erreurs pouvant être théoriquement corrigées). Cela signifie que chaque séquence RS est composée de 15 symboles de 4 bits, parmi lesquels 3 ( $=n-2t$ ) portent des données. En d'autres termes, 12 bits ( $3 \times 4$ ) sont disponibles pour définir le numéro du mode. En conséquence, le nombre de possibilités est de 4096 ( $2^{12}$ ).

Par ailleurs, même si le nombre maximum d'erreurs pouvant être corrigées est de 6, on limite à une seule correction, de façon à avoir une probabilité négligeable de fausse détection (par augmentation de la distance de Hamming entre n'importe quelle séquence aléatoire et les séquences sélectionnées).

Cependant, il a été conservativement supposé que deux RS ID pouvaient être envoyés successivement et que deux RS ID pouvaient aussi être envoyés sur des fréquences juxtaposées. Comme les fausses détections avec une partie de l'un et une partie de l'autre (que ce soit dans le domaine temps ou dans le domaine fréquence) doivent être évitées, il a été déterminé un sous-ensemble de RS ID qui sont réellement mutuellement indépendants ("orthogonaux"), c'est à dire que deux séquences RS ID contiguës ne peuvent produire un code RS ID valide mais faux. Ce sous-ensemble comprend seulement 272 possibilités. C'est le premier choix.

Chaque symbole est transmis dans une modulation MFSK. Il y a 16 possibilités de fréquences séparées de  $11025/1024=10,766$  Hz, chaque transmission de symbole se faisant sur une seule fréquence sur une durée égale à  $1024/11025 \times 1000=92,88$  ms. Les 15 symboles sont donc envoyés en l'espace de  $15 \times 1024/11025=1,393$  s.

Pour simplifier, le décodage est fait suivant un algorithme de type « force brute » où toutes les possibilités sont explorées et non suivant un classique algorithme de décodage Reed Solomon. Les éventuelles solutions trouvées sont hiérarchisées en fonction de leur distance (0 pour aucune erreur ou 1 pour une erreur) à une solution valide et en fonction d'un pseudo rapport signal sur bruit (puissance maximum/puissance moyenne). L'avantage est donnée à la solution ayant la distance la plus petite et le plus grand rapport signal sur bruit.

En outre, Vojtech (OK1IAK) a introduit une technique dite de « hachage » pour accélérer le calcul.

Pour chaque demi-pas de temps (46,44 ms) et pour chaque demi-pas de fréquence (5,38 Hz), le programme tente de détecter la présence d'un RS ID s'étendant sur les dernières 1,393 secondes. Donc, chaque seconde, environ 8500 possibilités de RS ID (en fonction de la largeur de bande choisie) sont testées. Comme la probabilité de fausse détection est presque nulle, il n'y a pas de problème à tester autant de possibilités.

L'analyse est basée sur des FFT (Fast Fourier Transform) de 2048 points à 11025 échantillons/sec, régulièrement faites à chaque demi-pas de temps (46,44 ms).

Pour la détection SdR, l'échantillonnage se fait à 44100 Hz à travers une décimation depuis 48, 96 ou 192 KHz. Elle permet de détecter n'importe quel RS ID dans une bande de 44 KHz.

C'est une transmission sans erreur, c'est à dire qu'en présence d'un identificateur RS ID, il y a deux solutions:

- \* soit l'identificateur n'est pas reçu parce que le signal est trop faible,
- \* soit il est reçu et alors il est correct, la probabilité de détection d'un mauvais identificateur RS ID étant quasiment nulle.

### Transmission

Dans presque tous les modes (y compris les modes MT63, SSTV, Fax, FELD HELL...), un RS ID peut être transmis. Il est envoyé à la fréquence moyenne de transmission, sauf en SSTV où il est envoyé sur le pic de synchronisation tel qu'affiché sur la « chute d'eau » (« waterfall »), c'est à dire entre 1150 et 1400 Hz.

### Relativement à la puissance moyenne transmise

Le rapport entre la puissance moyenne et la puissance maximale est égal à 1 car le RS ID est transmis en MFSK, donc la puissance transmise est maximum.

### Vidéo

La vidéo dont l'adresse WEB figure ci-dessous montre l'utilisation du RS ID sur Fldigi, Multipsk et DM780 (merci à Tony – K2MO)

<https://www.youtube.com/watch?v=nwkz0GNpA0I>

# L'identificateur de mode visuel (« ID Vidéo » ou « Video ID » en anglais)

## Origine

L'idée originale vient de Henri (F6BAZ), qui proposa (en décembre 2005) d'utiliser la « chute d'eau » pour afficher le nom du mode en CMT Hell. L'auteur (Patrick Lindecker) trouva l'idée intéressante et l'intégra à la version 3.12 de Multipsk (janvier 2006), pour tous les modes.

## Principale utilisation

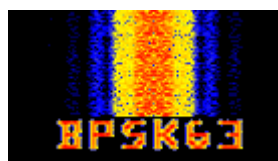
Il est envoyé en CMT Hell (Hellschreiber) le nom du mode et/ou d'autres informations avant la transmission principale. Cet identificateur sera visible sur la « chute d'eau » ("waterfall") de la station réceptrice.

Cet identificateur peut être transmis :

- soit sous forme horizontale : la transmission est rapide mais peu puissante,
- soit sous forme verticale : la transmission est lente mais plus puissante car plus concentrée (beaucoup moins de porteuses).

Par exemple, juste avant un appel général ("CQ") en BPSK63, "BPSK63" sera automatiquement transmis en CMT Hell. La station réceptrice verra "BPSK63" affiché sur sa « chute d'eau » et l'opérateur commutera immédiatement sur le mode BPSK63 (la commutation n'est pas automatique comme avec le RS ID).

Voici ce qui est vu sur la « chute d'eau » (ID Vidéo suivi de la transmission elle-même).



Pour l'instant, quatre programmes proposent le ID Vidéo (à ma connaissance): DM780, FIDigi, Digipan et Multipsk.

## Principe

Chaque caractère est transformé en majuscules, si besoin, puis en matrice de pixels suivant la police choisie (voir ci-après).

En dehors du MT63 (qui est un cas particulier), la transmission se fait à 11025 échantillons/seconde. La FFT se fait à partir de 4096 échantillons en utilisant 2048 échantillons du précédent tampon de FFT donc avec, en fait, 2048 nouveaux échantillons seulement. Ceci revient à doubler la vitesse d'affichage avec une légère dépendance entre pixels successifs. En conséquence, entre 2 pixels, on trouvera des intervalles de fréquence et de temps, respectivement, de:

\*  $11025/4096 = 2,691$  Hz,

\*  $2048/11025 = 0,1858$  seconde.

Les pixels sont transmis dans le mode C(Concurrent) MT Hell. Pour plus de détails, voir le site de Pascal, F1ULT:

<http://f1ult.free.fr/FUZZY/indexZL1BPU.htm>

Pour résumer, la transmission est faite sur de nombreuses porteuses juxtaposées, envoyées en parallèle. La présence ou l'absence d'une porteuse détermine soit une trace (un point) sur la « chute d'eau » soit l'absence de trace. Donc chaque ligne sera composée d'un ensemble de points et un ensemble de lignes constituera une ligne de caractères.

Nota: pour rester linéaire (sans saturation), chaque porteuse sera limitée en amplitude.

Le nom du mode est centré au milieu de la bande passante du mode (voir l'exemple portant sur le BPSK63, au-dessus).

Si la transmission est inversée (pour les modulations FSK, MFSK et QPSK), le nom l'est aussi (il sera alors écrit de droite à gauche). Il est rappelé que normalement, en HF, on transmet les digimodes en USB et non en LSB, et ceci indépendamment de la fréquence.

Les noms de modes seront envoyés dans leur appellation standard: "BPSK31" "BPSK63", "MFSK16", "PAX2", "FAX"..., s'ils sont sans ambiguïté.

Cependant, il existe des cas particuliers. Par exemple, le MT63 sera de type "MT63 bande entrelacement" avec pour la largeur de bande: "500", "1K" (pour "1000") ou "2K" (pour "2000"), et pour l'entrelacement: "VST" ("Very short" pour "Très court"), "ST" ("Short" pour "Court") et "LG" ("Long"), soit un maximum de 12 caractères pour, par exemple: "MT63 500 VST"

Voir l'aide Multipsk pour plus de détails sur la manière d'exprimer le nom des modes.

Les caractères peuvent être envoyés dans différentes polices Hellschreiber ("Hell 80 double", "Feld Hell double, normal", "Feld Hell double, gras").

#### Relativement à la puissance moyenne transmise

La puissance moyenne transmise est assez faible du fait qu'il est nécessaire de ne pas saturer le signal pour assurer la lisibilité du texte. Donc, plus le nombre de pixels "horizontaux" à transmettre sera grand et plus la puissance moyenne transmise sera faible. Cependant, la capacité "humaine" à lire un texte même très bruité compense, quelque peu, cette faible puissance.

# ANNEXE 1

## Liste des codes RS ID

"NUMBER" est le code RS ID. Les commentaires (en anglais) sont entre {}

```
IF NUMBER=1 THEN MODE:='BPSK31' ELSE
IF NUMBER=2 THEN MODE:='BPSK63' ELSE
IF NUMBER=3 THEN MODE:='QPSK63' ELSE
IF NUMBER=4 THEN MODE:='BPSK125' ELSE
IF NUMBER=5 THEN MODE:='QPSK125' ELSE
IF NUMBER=7 THEN MODE:='PSKFEC31' ELSE
IF NUMBER=8 THEN MODE:='PSK10' ELSE
```

{ "MT63" parameter 1 parameter 2 (parameter 1: "500", "1000" (1000 Hz) or "2000" (2000 Hz), parameter 2: "LG" (Long), "ST" (Short) or "VST" (Very short)  
Example: MT63-1000-LG}

```
IF NUMBER=9 THEN MODE:='MT63-500-LG' ELSE
IF NUMBER=10 THEN MODE:='MT63-500-ST' ELSE
IF NUMBER=11 THEN MODE:='MT63-500-VST' ELSE
IF NUMBER=12 THEN MODE:='MT63-1000-LG' ELSE
IF NUMBER=13 THEN MODE:='MT63-1000-ST' ELSE
IF NUMBER=14 THEN MODE:='MT63-1000-VST' ELSE
IF NUMBER=15 THEN MODE:='MT63-2000-LG' ELSE
IF NUMBER=17 THEN MODE:='MT63-2000-ST' ELSE
IF NUMBER=18 THEN MODE:='MT63-2000-VST' ELSE
```

```
IF NUMBER=19 THEN MODE:='PSKAM10' ELSE
IF NUMBER=20 THEN MODE:='PSKAM31' ELSE
IF NUMBER=21 THEN MODE:='PSKAM50' ELSE
IF NUMBER=22 THEN MODE:='PSK63F' ELSE
IF NUMBER=23 THEN MODE:='PSK220F' ELSE
```

{ "CHIP 64" parameter 1 (parameter 1: "64" or "128"), Example: CHIP-64}  
IF NUMBER=24 THEN MODE:='CHIP-64' ELSE  
IF NUMBER=25 THEN MODE:='CHIP-128' ELSE

```
IF NUMBER=26 THEN MODE:='CW' ELSE
```

{ "CCW" parameter 1 parameter 2 (parameter 1: "OOK" or "FSK",  
parameter 2: "12" (12 wpm), "24" (24 wpm) or "48" (48 wpm)  
Examples: CCW-OOK-12 or CCW-FSK-24}

```
IF NUMBER=27 THEN MODE:='CCW-OOK-12' ELSE
IF NUMBER=28 THEN MODE:='CCW-OOK-24' ELSE
IF NUMBER=29 THEN MODE:='CCW-OOK-48' ELSE
IF NUMBER=30 THEN MODE:='CCW-FSK-12' ELSE
IF NUMBER=31 THEN MODE:='CCW-FSK-24' ELSE
IF NUMBER=33 THEN MODE:='CCW-FSK-48' ELSE
```

{Factor1 ARQ not RX/TX in Multipsk 4.1.1}

```
IF NUMBER=34 THEN MODE:='FACTOR1-FEC' ELSE
```

{ "PACKET" parameter 1 (parameter 1: "300" (bauds) or "1200" (bauds)), Example:  
PACKET-300}

IF NUMBER=35 THEN MODE:='PACKET-300' ELSE  
IF NUMBER=36 THEN MODE:='PACKET-1200' ELSE

{"ASCII" parameter 1 (parameter 1: "7" (7 bits) or "8" (8 bits)), Example: ASCII-7}  
IF NUMBER=37 THEN MODE:='ASCII-7' ELSE  
IF NUMBER=38 THEN MODE:='ASCII-8' ELSE

{"RTTY" parameter 1 (parameter 1: "45" (45 bauds), "50" (50 bauds), "75" (75  
bauds)), Example: RTTY-45}  
IF NUMBER=39 THEN MODE:='RTTY-45' ELSE  
IF NUMBER=40 THEN MODE:='RTTY-50' ELSE  
IF NUMBER=41 THEN MODE:='RTTY-75' ELSE

IF NUMBER=42 THEN MODE:='AMTOR FEC' ELSE

{"THROB" parameter 1 (parameter 1: "1" (1 baud), "2" (2 bauds) or "4" (4 bauds)),  
Example: THROB-2}  
IF NUMBER=43 THEN MODE:='THROB-1' ELSE  
IF NUMBER=44 THEN MODE:='THROB-2' ELSE  
IF NUMBER=45 THEN MODE:='THROB-4' ELSE

{"THROBX" parameter 1 (parameter 1: "1" (1 baud) or "2" (2 bauds)), Example:  
THROBX-2}  
IF NUMBER=46 THEN MODE:='THROBX-1' ELSE  
IF NUMBER=47 THEN MODE:='THROBX-2' ELSE

{"CONTESTIA" parameter 1 parameter 2 (parameter 1: "4" (4 tones), "8" (8 tones),  
"16" (16 tones), "32" (32 tones), parameter 2: "250" (B=250 Hz), "500" (B=500 Hz) or  
"1000" (B=1000 Hz))...same parameters as OLIVIA

Examples: CONTESTIA-32-1000 or CONTESTIA-8-500

Note: the following are the main Contestia modes:

CONTESTIA-4-250, CONTESTIA-4-500, CONTESTIA-8-250, CONTESTIA-8-500,  
CONTESTIA-16-500, CONTESTIA-16-1000, CONTESTIA-32-1000}

IF NUMBER=49 THEN MODE:='CONTESTIA-8-250' ELSE  
IF NUMBER=50 THEN MODE:='CONTESTIA-16-500' ELSE  
IF NUMBER=51 THEN MODE:='CONTESTIA-32-1000' ELSE  
IF NUMBER=52 THEN MODE:='CONTESTIA-8-500' ELSE  
IF NUMBER=53 THEN MODE:='CONTESTIA-16-1000' ELSE  
IF NUMBER=54 THEN MODE:='CONTESTIA-4-500' ELSE  
IF NUMBER=55 THEN MODE:='CONTESTIA-4-250' ELSE

IF NUMBER=56 THEN MODE:='VOICE' ELSE  
IF NUMBER=57 THEN MODE:='MFSK16' ELSE  
IF NUMBER=60 THEN MODE:='MFSK8' ELSE

{"RTTYM" parameter 1 parameter 2 (parameter 1: "4" (4 tones), "8" (8 tones), "16"  
(16 tones), "32" (32 tones), parameter 2: "250" (B=250 Hz), "500" (B=500 Hz) or  
"1000" (B=1000 Hz))...same parameters as OLIVIA

Examples: RTTYM-32-1000 or RTTYM-8-500

Note: the following are the main RTTYM modes: RTTYM-4-250, RTTYM-4-500,  
RTTYM-8-250, RTTYM-8-500, RTTYM-16-500, RTTYM-16-1000, RTTYM-32-1000}

IF NUMBER=61 THEN MODE:='RTTYM-8-250' ELSE  
IF NUMBER=62 THEN MODE:='RTTYM-16-500' ELSE  
IF NUMBER=63 THEN MODE:='RTTYM-32-1000' ELSE

```
IF NUMBER=65 THEN MODE:='RTTYM-8-500' ELSE
IF NUMBER=66 THEN MODE:='RTTYM-16-1000' ELSE
IF NUMBER=67 THEN MODE:='RTTYM-4-500' ELSE
IF NUMBER=68 THEN MODE:='RTTYM-4-250' ELSE
```

{ "OLIVIA" parameter 1 parameter 2 (parameter 1: "4" (4 tones), "8" (8 tones), "16" (16 tones), "32" (32 tones), parameter 2: "250" (B=250 Hz), "500" (B=500 Hz) or "1000" (B=1000 Hz))

Examples: OLIVIA-32-1000 or OLIVIA-8-500

Note: the following are the main Olivia modes: OLIVIA-4-250, OLIVIA-4-500, OLIVIA-8-250, OLIVIA-8-500, OLIVIA-16-500, OLIVIA-16-1000, OLIVIA-32-1000}

```
IF NUMBER=69 THEN MODE:='OLIVIA-8-250' ELSE
IF NUMBER=70 THEN MODE:='OLIVIA-16-500' ELSE
IF NUMBER=71 THEN MODE:='OLIVIA-32-1000' ELSE
IF NUMBER=72 THEN MODE:='OLIVIA-8-500' ELSE
IF NUMBER=73 THEN MODE:='OLIVIA-16-1000' ELSE
IF NUMBER=74 THEN MODE:='OLIVIA-4-500' ELSE
IF NUMBER=75 THEN MODE:='OLIVIA-4-250' ELSE
```

```
IF NUMBER=76 THEN MODE:='PAX' ELSE
IF NUMBER=77 THEN MODE:='PAX2' ELSE
IF NUMBER=78 THEN MODE:='DOMINOF' ELSE
IF NUMBER=79 THEN MODE:='FAX' ELSE
```

{the sub-modes SSTV are automatically recognized in SSTV so no parameters are needed for SSTV}

```
IF NUMBER=81 THEN MODE:='SSTV' ELSE
```

{ "DOMINOEX" parameter 1 and optionally parameter 2 (parameter 1: "4" (4 baud) , "5" (5 bauds), "8" (8 bauds), "11" (11 bauds), "16" (16 bauds), "22" (22 bauds), parameter 2 "" (no parameter 2 for non-FEC) or "FEC" (for FEC correction))

Example: DOMINOEX-11 or DOMINOEX-16-FEC (parameter 2 is optional)}

```
IF NUMBER=84 THEN MODE:='DOMINOEX-4' ELSE
IF NUMBER=85 THEN MODE:='DOMINOEX-5' ELSE
IF NUMBER=86 THEN MODE:='DOMINOEX-8' ELSE
IF NUMBER=87 THEN MODE:='DOMINOEX-11' ELSE
IF NUMBER=88 THEN MODE:='DOMINOEX-16' ELSE
IF NUMBER=90 THEN MODE:='DOMINOEX-22' ELSE
IF NUMBER=92 THEN MODE:='DOMINOEX-4-FEC' ELSE
IF NUMBER=93 THEN MODE:='DOMINOEX-5-FEC' ELSE
IF NUMBER=97 THEN MODE:='DOMINOEX-8-FEC' ELSE
IF NUMBER=98 THEN MODE:='DOMINOEX-11-FEC' ELSE
IF NUMBER=99 THEN MODE:='DOMINOEX-16-FEC' ELSE
IF NUMBER=101 THEN MODE:='DOMINOEX-22-FEC' ELSE
```

```
IF NUMBER=104 THEN MODE:='FELD HELL' ELSE
IF NUMBER=105 THEN MODE:='PSK HELL' ELSE
IF NUMBER=106 THEN MODE:='HELL 80' ELSE
```

{ "FMHELL" parameter 1 (parameter 1: "105" (105 bauds) or "245" (245 bauds)),

Example: FMHELL-245}

```
IF NUMBER=107 THEN MODE:='FM HELL-105' ELSE
IF NUMBER=108 THEN MODE:='FM HELL-245' ELSE
```

```
IF NUMBER=110 THEN MODE:='QPSK31' ELSE
```

```
{"PACKET" parameter 1 (parameter 1: "110" (bauds),Example: PACKET-110}  
IF NUMBER=113 THEN MODE:='PACKET-110' ELSE  
  
IF NUMBER=114 THEN MODE:='141A' ELSE  
  
IF NUMBER=116 THEN MODE:='OLIVIA-8-1000' ELSE  
IF NUMBER=117 THEN MODE:='CONTESTIA-8-1000' ELSE  
IF NUMBER=119 THEN MODE:='RTTYM-8-1000' ELSE  
  
IF NUMBER=123 THEN MODE:='DTMF' ELSE  
  
IF NUMBER=125 THEN MODE:='ALE400' ELSE  
  
IF NUMBER=126 THEN MODE:='BPSK250' ELSE  
IF NUMBER=127 THEN MODE:='QPSK250' ELSE  
  
IF NUMBER=131 THEN MODE:='FDMDV' ELSE  
  
IF NUMBER=132 THEN MODE:='JT65-A' ELSE  
IF NUMBER=134 THEN MODE:='JT65-B' ELSE  
IF NUMBER=135 THEN MODE:='JT65-C' ELSE  
  
IF NUMBER=136 THEN MODE:='THOR-4' ELSE  
IF NUMBER=137 THEN MODE:='THOR-8' ELSE  
IF NUMBER=138 THEN MODE:='THOR-16' ELSE  
IF NUMBER=139 THEN MODE:='THOR-5' ELSE  
IF NUMBER=143 THEN MODE:='THOR-11' ELSE  
IF NUMBER=145 THEN MODE:='THOR-22' ELSE  
  
IF NUMBER=146 THEN MODE:='THROBX-4' ELSE  
  
IF NUMBER=147 THEN MODE:='MFSK32' ELSE  
IF NUMBER=148 THEN MODE:='MFSK11' ELSE  
IF NUMBER=152 THEN MODE:='MFSK22' ELSE  
  
IF NUMBER=153 THEN MODE:='CALL ID' ELSE  
  
{"PACKET" parameter 1 (parameter 1: "PSKbauds")}  
IF NUMBER=155 THEN MODE:='PACKET-PSK1200' ELSE  
IF NUMBER=156 THEN MODE:='PACKET-PSK250' ELSE  
IF NUMBER=159 THEN MODE:='PACKET-PSK63' ELSE  
  
IF NUMBER=163 THEN MODE:='OLIVIA-8-125' ELSE  
IF NUMBER=169 THEN MODE:='CONTESTIA-8-125' ELSE  
IF NUMBER=170 THEN MODE:='RTTYM-8-125' ELSE  
  
{MIL-STD-188-110A in asynchronous format 8N1 (8 data bits, no parity, 1 stop bit)}  
IF NUMBER=172 THEN MODE:='110A-8N1' ELSE  
  
IF NUMBER=173 THEN MODE:='BPSK500' ELSE  
  
{ "PSK Robust" or "PSKR" in short}  
IF NUMBER=183 THEN MODE:='PSK125R' ELSE
```

```
IF NUMBER=186 THEN MODE:='PSK250R' ELSE
IF NUMBER=187 THEN MODE:='PSK500R' ELSE
```

```
{"PACKET" PSK at 31 bauds (parameter 1: "PSKbauds")}
IF NUMBER=189 THEN MODE:='PACKET-PSK31' ELSE
```

```
{Contestia modes for UHF and emergency communications }
IF NUMBER=191 THEN MODE:='CONTESTIA-64-2000' ELSE
IF NUMBER=193 THEN MODE:='CONTESTIA-64-1000' ELSE
IF NUMBER=194 THEN MODE:='CONTESTIA-64-500' ELSE
IF NUMBER=201 THEN MODE:='CONTESTIA-32-2000' ELSE
```

```
{additive Olivia and Contestia modes}
IF NUMBER=203 THEN MODE:='OLIVIA-4-125' ELSE
IF NUMBER=204 THEN MODE:='CONTESTIA-4-125' ELSE
IF NUMBER=211 THEN MODE:='OLIVIA-64-2000' ELSE
IF NUMBER=214 THEN MODE:='OLIVIA-8-2000' ELSE
IF NUMBER=221 THEN MODE:='OLIVIA-32-2000' ELSE
IF NUMBER=229 THEN MODE:='OLIVIA-4-1000' ELSE
IF NUMBER=234 THEN MODE:='OLIVIA-16-2000' ELSE
IF NUMBER=238 THEN MODE:='OLIVIA-4-2000' ELSE
IF NUMBER=247 THEN MODE:='CONTESTIA-8-2000' ELSE
IF NUMBER=254 THEN MODE:='CONTESTIA-4-2000' ELSE
IF NUMBER=255 THEN MODE:='CONTESTIA-4-1000' ELSE
IF NUMBER=259 THEN MODE:='CONTESTIA-16-2000' ELSE
IF NUMBER=261 THEN MODE:='CONTESTIA-128-2000' ELSE
```

```
{mode Lentus}
IF NUMBER=262 THEN MODE:='LENTUS' ELSE
```

```
{Extension of RSID, for a two sequences RSID code}
IF NUMBER=6 THEN MODE:='EXTENSION_FLDIGI';
```

Nota : certains modes Fldigi sont définis par une double séquence de codes. Par exemple, le « MFSK64 » est défini par le code « 6 » suivi du code « 620 »