

Technical aspects of RS ID and Call ID and use

Patrick Lindecker (F6CTE) the 15th of July 2010
Thanks to Bill Duffy (KA0VXK) for proofreading this text

Summary

1. Why use RS ID
2. RS ID origin, main use and maintenance
3. Description of RS ID
4. Main use of Call ID
5. Description of Call ID

Appendix 1 : List of the RS ID codes.

1. Why use RS ID

The identification of a digital amateur mode (commonly called “digimode”) can't be automatically done, in general, by software. This is due to the diversity of the modulations used (BPSK, QPSK, FSK, MFSK, IFK, OOK...), speeds used (from 1 to 9600 bauds), codings used (convolutional, Reed Solomon or other). Also, it may have several transmissions of different types on the same bandwidth, which makes the problem even more difficult.

However, some modes are simple to identify by the user, either because their frequency is well known (BPSK31 on 14070 KHz, for example) or because the visual signature on the waterfall or the acoustic signature is characteristic (RTTY 45 baud for example or CW).

But for the other digimodes (more or less exotic), it is very difficult to identify the mode or the sub-mode used, with the simple visual and/or acoustic traces. Therefore the need of digimodes identifiers.

At the present time, the only official identifier is the one which defines the analog SSTV sub mode used (Robot 36, Martin 1...).

In general, it is possible to send a small CW text before each digimode transmission. But this is rarely used.

A more modern way to signal the mode is the Video ID where the mode is directly displayed on the waterfall (proposed by DM780, FIDigi, Digipan and Multipsk).



However, due to the transmission mode (MT Hell), the transmitted power is weak. Consequently, the transmission range is weak also.

Over, these last years a lot of amateur modes and sub-modes were created. For example: there exists a mode called Olivia from which has been derived several other modes: Contestia, RTTYM, Pax, Pax2 and Voice. Besides, Olivia, Contestia and RTTYM have each one 40 sub-modes (of course, not all used). Without any way to discriminate modes and sub-modes, it is almost impossible to know the used mode/sub-mode, which would be very frustrating and could discourage the use of digital modes.

Another problem which frequently occurs is the need of a precise tuning, for example with MFSK or THROB modes. The tuning can last some time and can cause the loss of a lot of decoded

characters. This can also lead to a non-established QSO. So an automatic tuning is also requested.

The RS ID answers these two problems (detection of the mode used and automatic tuning).

RS ID is also now used for automatic switching for quick adaptation to the QSO conditions in Fldigi and for automatic re-tuning in ALE400 ARQ FAE in case of loss of link (Multipsk).

2. RS ID origin, main use and maintenance

The "RS ID" ("RS" for "Reed-Solomon" and « ID » for « Identifier ») was created by Patrick Lindecker, F6CTE and added to the version 4.1.1 of Multipsk (November 2006).

The RS ID allows the automatic identification of any digital transmission done in one of 143 (at the time of this paper) RX/TX modes handled by different decoding softwares (PocketDigi, FDMDV, DM780, Fldigi, Multipsk). None of this software support all the modes. Each decoding program decodes a subset of the set of modes detected through the RS ID.

However, Fldigi, Multipsk...announce all the detected modes even if they are not decoded.

On reception of a RS ID, two events occur : the mode used is detected and the central frequency of the RS ID, which is also the central frequency of the identified mode, is determined with a precision of 2.7 Hz. This is sufficient to allow all current modes to begin accurate decoding. This is an excellent way to insure that signals like MFSK are properly tuned and decoded.

This RS ID signal is transmitted in 1.4 sec and has a bandwidth of 172 Hz. Detection of the RS ID signal is possible down to a Signal to Noise ratio of about -16 dB, so with a sensitivity equal or better than the majority of the digital modes (RTTY, PSK31...), except several modes as PSK10, PSKAM10, THROB, THROBX or JT65.

Note: consequently, it is possible to detect RS ID and not be able to decode the ensuing data signal because it is too weak.

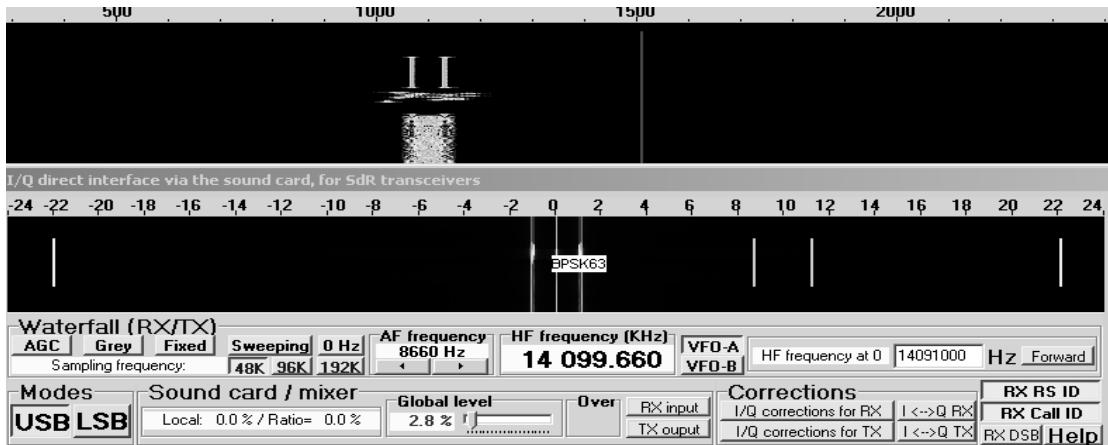
The RS ID is normally transmitted in a leading position, i.e. before each general call or prior to each answer in a QSO. However, it can also been transmitted at the end of each transmission.

The search is, in general, done in the entire audio spectrum (several KHz). For this bandwidth, the equivalent CPU load is negligible. But the search can be limited in a narrow bandwidth centered on the current audio subcarrier.

It can also be done on a 44 KHz bandwidth from a SdR receiver. In that case, the equivalent CPU load is about 500 MHz (on Multipsk). Below is a snapshot of the SdR Multipsk interface on reception of a RS ID.

Note 1: RS ID detection occurs as a background process and does not interfer with the normal signal decoding.

Note 2: false detection is possible but statistically rare (never seen by the author even on 44 KHz bandwidth), this is due to the use of a very strong autocorrelation function associated with the RS ID codes.



At the present time, five programs offer the RS ID:

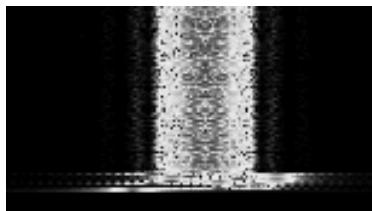
- PocketDigi by Vojtech, OK1IAK
- FDMDV by Cesco, HB9TLK
- DM780 by Simon, HB9DRV
- fldigi by Dave, W1HKJ
- Multipsk by Patrick, F6CTE

RS ID programs sources are available either, in C++, from Vojtech (OK1IAK) or from the author (in Delphi and including SdR detection).

The master list of the RS ID code/mode combinations is maintained by the author, F6CTE (in order to maintain compatibility between the programs supporting RS ID) and changes made to the list in concurrence with the above list of software developers. This list is distributed on the DigitalRadio Yahoo group at each change.

A mode is integrated if a real interest for Hams can be shown. All submodes or variations around a mode (shift for example) can't be integrated as there is a limited number of possible RS ID codes.

For example, here is the PSK63 RS ID (root) received just before the PSK63 transmission itself (trunk).



3. Description of the RS ID

Each mode corresponds to a number which is transformed in a particular Reed-Solomon sequence. This RS coding (RS($k=4, t=6$)) is defined by the parameters:

- $k=4$ (number of bits per symbol),
- $n=15$ ($=2^k-1$, number of symbols by RS sequence) and
- $t=6$ (maximum number of errors which could be theoretically corrected).

Each RS sequence is composed of 15 symbols of 4 bits, among which 3 ($=n-2t$) contain data. In other words, 12 bits (3×4) are available to define the mode number. Consequently, the number of possibilities would be equal to 4096 (2^{12}).

In addition, even if the maximum number of errors which could be fixed is equal to 6, it is limited to one correction so as to have a negligible probability of false detection detection (because the Hamming distance between any random sequence and the selected sequences increases).

It has been conservatively supposed that two RS ID could be sent sequentially and that two RS ID could be also transmitted in juxtaposed frequencies. As false RS ID detection with part of one and part of the other (either in the time domain or the frequency domain) must be avoided, it is necessary to identify a subset of the 4096 RS ID codes that are mutually independent ("orthogonal"). That is, two contiguous RS ID sequences can't produce a valid but incorrect RS ID code. This subset is composed of only 272 unique values.

Other codes can be assigned temporarily for use during the development of a new modem type.

If every known digital signal were assigned a unique code, the list of 272 would quickly be exhausted. It is for that reason that not every combination of baud rate, tone numbers and bandwidth for modes as RTTY...has a specific RS ID.

Each symbol is transmitted using MFSK modulation. There are 16 possibilities of frequencies separated by $11025/1024=10.766$ Hz. Each symbol transmission is done on only one frequency for a duration equal to $1024/11025 \times 1000=92.88$ ms. So, the entire RS ID sequence of 15 symbols is transmitted in $15 \times 1024/11025=1.393$ s. The RS ID signal is transmitted at the maximum peak power of the associated modem signal.

Decoding can be accomplished by a " brute force " algorithm where all the possibilities are tested (and not according to a classical Reed Solomon decoding algorithm). The possible found solutions are sorted according to their distance (0 error or 1 error) to a valid solution and according to a pseudo signal-to-noise ratio (peak power/average power). The final detected code will have the lowest distance with the biggest pseudo signal-to-noise ratio.

Vojtech, OK1IAK, developed a " hashing " algorithm which reduces the detector processing load. He kindly made that code available to other developers. Both fldigi and DM780 use Vojtech's decoding algorithm. RS ID in Multipsk also uses a hashing algorithm (but different).

For each semi-step of time (46.44 ms) and for each semi-step of frequency (5.38 Hz), the program attempts to detect a RS ID extending for the last 1.393 seconds. So each second, about 8500 possible RS ID (depending on the selected bandwidth) are tested (depending on the bandwidth).

The analysis is based on FFT (Fast Fourier transform) of 2048 points at 11025 samples/sec, regularly done at each semi-step of time (46.44 ms).

For SdR detection, the sampling is done at 44100 Hz, through a decimation from 48 KHz, 96 KHz or 192 KHz. It permits to detect any RS ID within a bandwidth of 44 KHz.

In presence of a RS ID identifier, there are only two possible outputs from the detection algorithm:

- * either the RS ID identifier is not received because the signal is too weak,
- * or it is received and it is correct, the probability of detecting a wrong RS ID being almost nil.

Note about the transmission frequency

In all modes (including MT63, SSTV, Fax, FELD HELL...), a RS ID can be transmitted. It is sent on the average transmission frequency, except in SSTV where it is sent on the synchronization peak as displayed on the waterfall (between 1150 and 1400 Hz).

Note about the average transmitted power

The ratio between the average power and the maximal power is equal to 1 because the RS ID is transmitted in MFSK, so the transmitted power is maximum.

Note about the list of RS ID codes

A list of the RS ID codes (the last one distributed to the digital Yahoo group) is proposed in Appendix 1.

Example of RS ID use

Based on snapshots, this paper shows how to use easily RS ID with Multipsk :
« http://f6cte.free.fr/The_RS_ID_easy_with_Multipsk.doc »

4. Main use of Call ID

The Call ID ("Call" for "Call sign" and « ID » for « Identifier ») is an extension of a particular RS ID (code 153). It was created by Patrick Lindecker, F6CTE and added to the version 4.12 of Multipsk (December 2008).

This identifier is transmitted in 3.3 sec and has a bandwidth of 172 Hz. It is composed of a RS ID (code 153) followed by a sequence of 56 bits of data and 24 bits of CRC, which permits additional information.

The "Call ID" identifier can be, automatically, detected and located. It can be :

- any Call sign (formatted or free),
- a Call sign 6 characters + Locator 6 characters ,
- a « Prop ID » (« Prop » for « Propagation ») : Call sign 6 characters + Locator 4 characters « + » + power level, antenna type and directivity,
- a Message ID : message of 9 characters maximum.

The Call ID is always detected at -13 dB (first decodings at -18 dB), so with a sensitivity better than PSK31. The central frequency of the identifier (which is also the central frequency of the transmission), is determined with a precision of +/- 2.7 Hz.

As soon as this identifier is received, Multipsk:

- either displays the formatted call sign (possibly preceded by "CQ " and followed by "Emergency!" (and a short ring))
or displays the free call sign,
or displays the call sign (possibly preceded by "CQ ") and the Locator 6 characters on the waterfall,
or displays the call sign (possibly preceded by "CQ "), the Locator 4 characters « + », the power level (in Watts), antenna type or gain and directivity of the antenna,
- displays the time of reception, the same pieces of information as above and the Signal to Noise ratio in the Call ID window,
- switches on the received frequency (optional),
- displays the position of the Locator on a map (optional).

At the present time, only Multipsk offers the Call ID.

Call ID program source is available from the author (in Delphi).

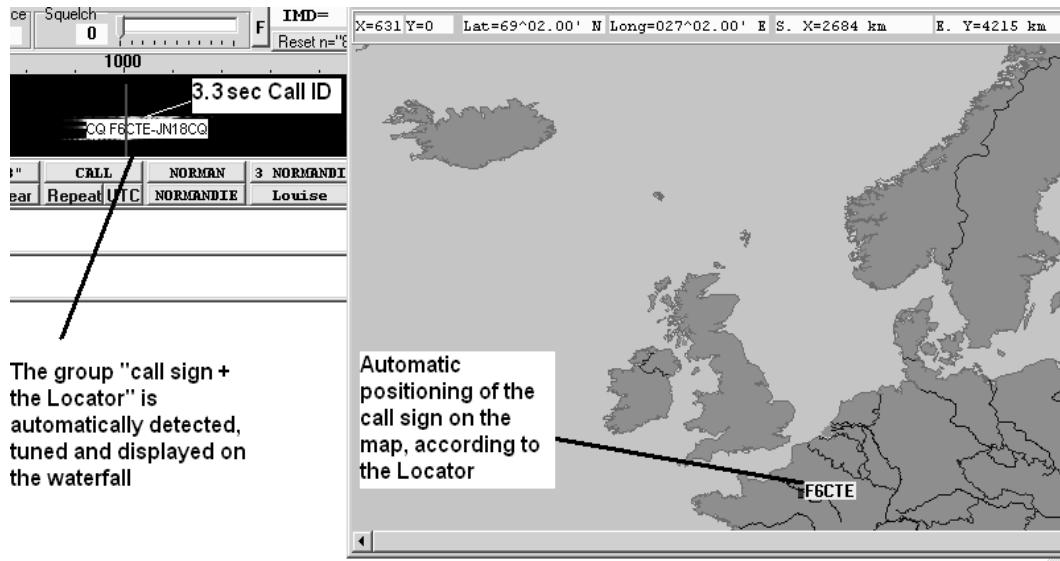
As for RS ID, in presence of a Call ID identifier, there are only two possible outputs from the detection algorithm:

- * either the Call ID identifier is not received because the signal is too weak,
- * or it is received and it is correct. The probability of detecting a wrong Call ID is almost nil (due to the CRC).

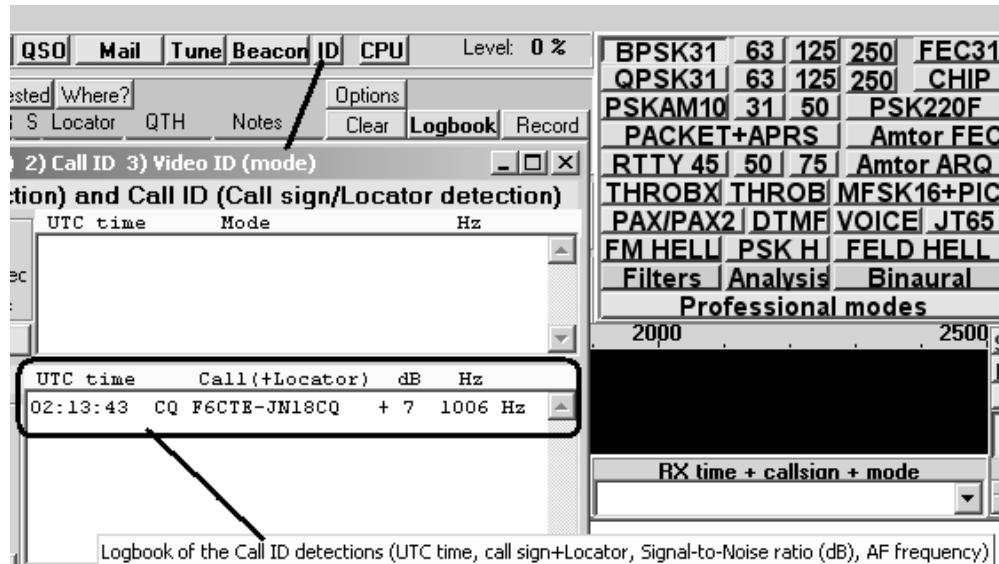
Example of Call ID use

Based on snapshots, this paper shows how to easily use Call ID with Multipsk :
[« http://f6cte.free.fr/The_Call_ID_and_Prop_ID_easy_with_Multipsk.doc »](http://f6cte.free.fr/The_Call_ID_and_Prop_ID_easy_with_Multipsk.doc)

Example of reception of a Call ID (call sign + Locator)



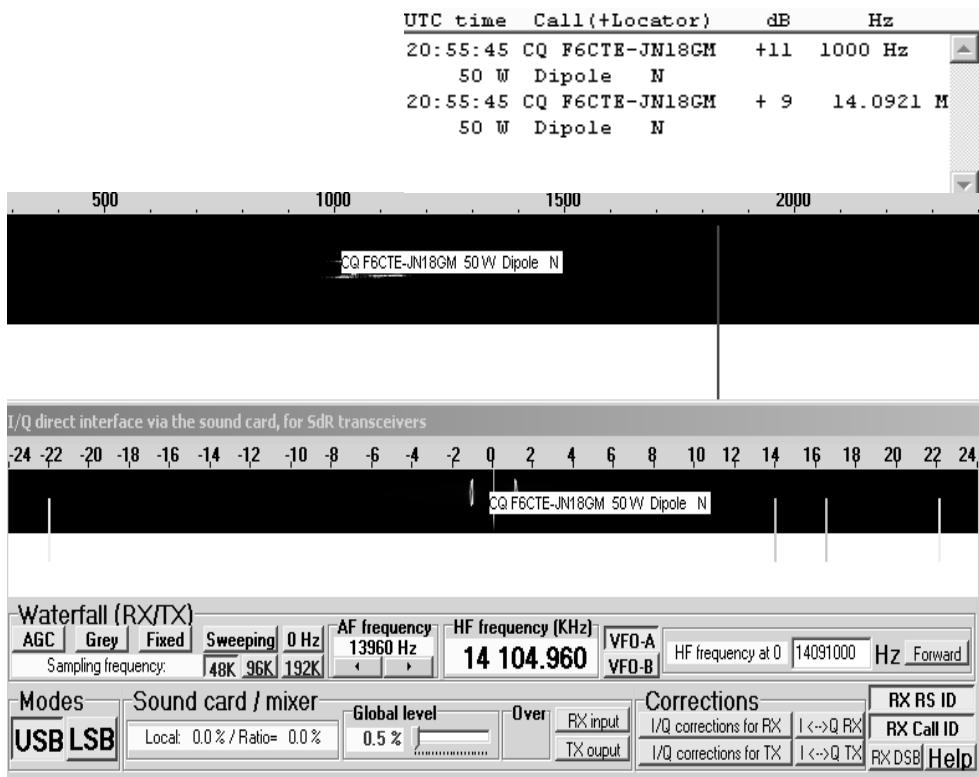
Log book of the Call Ids



Call ID (Prop ID in the example below) and SdR reception

From the Multipsk SdR window ("Direct via the sound card" button), it is proposed to detect any RS ID, Call ID (Prop ID here) on a 44 KHz bandwidth (between the yellow vertical dashes) and to display the RS ID or the Call ID in the SdR spectrum.

For example, with a SdR receiver tuned to a central frequency of 14.091 MHz, it will be possible to detect any Call ID on all the 14 MHz digital band, i.e. from 14.070 to 14.112 MHz. Below



Message ID

It is possible to send a message of 9 characters maximum in background. It is not related to any specific mode. Message IDs are not considered as true Call ID and are not stored.

The main use is simply to send a short message in case of difficulty during a QSO ("PSE PSK10", for example, to ask "To switch to PSK10" or "QRM +1K" for "There is QRM, I increase the dial frequency by 1 KHz" or "STIL HERE" to say that "I'm still here, even if communication seems impossible".

It will be efficient to use Q code and Ham abbreviations and to use punctuations to limit the number of characters:

- "?" question (a confirmation is required),
- "!" a strong request is issued by the Ham with whom you are in QSO (an action is required),
- ">" or "-" for "to increase the frequency",
- "<" or "-" for "to decrease the frequency",
- "=" at the beginning without following space for "PSE " or "Please "
- "+" at the end of the message for "I wait for an answer"
- and why not net general smileys as ":-)" for "Smile"

Example of message ID « HELLO ALL »



5. Description of the Call ID

GENERALITIES

This ID sequence is composed of 80 bits i.e 20 symbols of 4 bits transmitted as in RS ID transmissions, divided into:

- 56 bits of data
- 16 bits of CRC (same CRC as in Packet)
- 8 first bits of a CRC (same CRC as in Packet)

This sequence is preceded by a specific RS ID (CALL ID: Code 153)

The CALL ID is immediately transmitted after the RS ID

The transmission of the CALL ID symbols are done exactly as the RS ID symbols and in the same frequency.

Call ID Transmission part:

This Call ID is transformed in a 56 bits sequence. The 24 bits of CRC is added.

These 80 bits are formed into 20 symbols of 4 bits (as for RS ID) (number 0 to 19).

As for RS ID, each symbol corresponds to a frequency (one among 16), 0 corresponds to the first (the lowest) frequency and 15 to the last (the highest) frequency. The first symbol (n°0) is the first to send and the last (n°19) is the last to send.

Carriers are separated by a difference of $11025/1024=10.766$ Hz, so the bandwidth of the identifier is about $16 \times 10.766 = 172$ Hz.

The central frequency and the 16 frequencies are the same as the ones used for the RS ID transmission.

Symbols are separated by $1024/11025$ sec so an identifier lasts $20 \times 1024/11025 = 1.86$ sec.

Different types of Call ID

Taking advantage of the 56 data bits, different services have been developed. These are as follows :

1. the **standard Call ID** whose goal is mainly to send a call sign and, possibly a Locator.
There are 3 subtypes of standard Call ID :
 - formatted call sign (14 characters maximum),
 - free call sign (10 characters maximum),
 - call sign (6 characters maximum) + Locator (6 characters)
2. the **Prop ID** composed of a call sign (6 characters maximum) + a Locator (4 characters « + ») + the power level (in Watts), antenna type or gain and directivity of the antenna,
3. the **message ID** (9 characters maximum).

These different types of Call ID are described below.

DESCRIPTION OF THE STANDARD CALL ID

For the standard Call ID, 3 possible call sign types can be transmitted:

FORMATTED CALL SIGNS

The user gives a valid call sign (even with foreign country call sign prefix and with Postfix as for example "AZ2/F6CTE/MM").

Ham Call signs: it is taken into account the World Radio Conference 2003 revised Article 19 (Call Sign Formulation) of the ITU Radio Regulations

Call signs are in the shape "Foreign prefix/Prefix+Mantissa/Suffix" as for example

"AZ2/WB4QSDA/MM" or

"AZ2/WB4123A/AM" Only Prefix and Mantissa are compulsory.

"C" for letters and figures,

"L" for capital letters (A to Z) and "F" for figures (0 to 9):

- Formation of the Prefix: LF/LLF/LFF/FLF or LF/LCF/FLF
- Formation of the standard Mantissa: L/CL/CCL/CCCL
- Formation of the Foreign Prefix: L/LL/LF/FL/LLF/LFF/FLF or L/LC/FL/LCF/FLF
- Formation of the Suffix: it can be M for mobile, P for portable operation, MM for maritime mobile or /AM (aeronautical mobile).

/A and /AM are considered as having the same meaning

To this formatted call sign can be added a bit of emergency:

Bit Emergency (FALSE (0) or TRUE (1)) to make know that the transmission of the formatted call sign is designed for emergency needs (high priority).

Notes:

* 7 figures are possible with calls: FLF/FLFCCCL

* 14 characters maximum LLF/LLFCCCL/MM

FREE CALL SIGNS (with bit CQ=FALSE)

It is proposed another way to form free call signs with 10 characters (A..Z, 0..9, space and /)

For example, "CQ F6/CTE1" would be accepted

CALL SIGN + LOCATOR 6 characters (Exemple: FW9XYZ+JN18CQ)

6 characters max standard call signs

- Formation of the Prefix: LF/LLF/LFF/FLF or LF/LCF/FLF
 - Formation of the standard Mantissa: L/CL/CCL
- Locator 6 characters (Exemple JN18CQ)
AA to RR / 00 to 99 / AA to XX

DESCRIPTION OF THE "PROP ID" (for Propagation ID)

CALL SIGN + reduced LOCATOR + Power level (Watt)+ gain of antenna + Directivity (Exemple: FW9XYZ+JN18GM+ 100 W +4 dB + South West)

6 characters max standard call signs

- Formation of the Prefix: LF/LLF/LFF/FLF or LF/LCF/FLF
- Formation of the standard Mantissa: L/CL/CCL

Locator 4 characters extended of one bit to have squares of 1 degree of longitude x 1 degree of latitude (Exemple JN18GM)

AA to RR / 00 to 99 / GM or SM (longitude and latitude precision= +/- 0.5 degree = +/- 55 kms = +/- 34 miles

Power level in Watts :

N	Watts
0	0.5
1	1
2	2
3	5
4	10
5	20
6	50
7	100
8	200
9	500

Relative gain of HF antenna with respect to isotropic antenna (same power in all directions):

N | Gain

0 | Vertical (omni directionnal)

1 | Dipole (gain around 2 dBi with a forward and backward radiation),

- 2 | Bidirectional antennas as colinear or group of parallel antennas (with a foreward and backward radiation), gain close to 5 or 6 dBi
- 2 | 4 dBi (for example Quad loop)
- 3 | 7 dBi (for example 2 elements Yagi)
- 4 | 10 dBi (for example 2 elements Quad or 3 elements Yagi)
- 5 | 13 dBi (for example 4 to 7 elements Yagi or 3 elements Quad)
- 6 | 16 dBi (for example 5 elements Quad)
- 7 | 19 dBi (for example multi elements Yagi or Quad)

4 to 19 dBi are related to monodirectional HF antennas as Yagi or Quad antenna with a backward radiated power being very reduced.

Directivity for all antennas except the vertical:

- N | Directivity
- 0 | North East
- 1 | East
- 2 | South East
- 3 | South
- 4 | South West
- 5 | West
- 6 | North West
- 7 | North

DESCRIPTION OF THE MESSAGE ID (with bit CQ=TRUE)

It is proposed to send a message with 9 characters maximum, with the following set of characters (56)

A..Z 0..9 space . ? = / + ! - , ; : % < > \$ " () * '

MISCELLANEOUS

Management of the Bit CQ (FALSE (0) or TRUE (1))

To make know that the transmission of the call sign is done to initiate a QSO or to answer to a call. Without CQ bit it means that it is a one way beacon (for propagation etc)

For FREE CALL SIGNS Bit CQ is always set to FALSE ("CQ" can be integered in the free call sign in needed)

For MESSAGE ID Bit CQ is always set to TRUE

Correction of wrong symbol

One symbol of 4 bits over 20 can be fixed

APPENDIX 1

List of the RS ID codes

"NUMBER" is worth for "code". Between {} are comments

```
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
```

{Pactor1 ARQ not RX/TX in Multipsk 4.1.1}

```
IF NUMBER=34 THEN MODE:='PACTOR1-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';
```