DCS “Tones” Low Bandwidth Coding

A Simple Coding for Data Transit

The DCS code set provides a low bandwidth squelch system for radio. Adapting this for data transmission has a number of benefits. Fast channel open by repetition of a specific code. Good pretested hardiness to transmission noise. A simply available decoder. Non interruption of FM voice by narrow bandwidth squelsh misfiring. The assignment of letter to code to provide for data communication is easily possible in the 83 standard codes, and as it appears 117 codes are possible, there is room for expansion. Maybe the squelch close code is valid, or invalid from the 117.

It is also possible to not use it on radio, and just use it locally at a local music venue.

As UHF is spread at 25kHz, and tuning is usually 12.5kHz, that an interband space with low interference can happen by placing transmission offset by 12.5kHz from a band centre. The FM locking effect will likely reduce interference, and advance recovery algorithms will assist in decode. The rigidity to noise of the (23, 11) field used in code production, and the very low bandwidth needed provides for excellent coding density and resilience.

The data rate is limited by the squelch coding symbol send rate, but this far exceeds typing speeds of the population. Preparation of content to send is a differential of online at the moment building of a sendable. Obviously a wide band upscale (perhaps by frequency multiplication) would be required for broadband (obvious, and some see the mini as fool, and brag of people they know not or want to flatter), but this should not preclude an ability at a low rate.

023@	114N	205***r+***	306***lf***	411***0***	503:	703***sp***
025A	115O	223***r-***	311'	412***1***	506;	712!
026B	116P	226***g+***	315(	413***2***	516<	723"
031C	125Q	243***g-***	331)	423***3***	532=	731£
032D	131R	244***b+***	343+	431***4***	546>	732$
043E	132S	245***b-***	346,	432***5***	565?	734%
047F	134T	251***up***	351-	445***6***		743^
051G	143U	261***dn***	364.	464***7***		754&
054H	152V	263***le***	365/	465***8***		
065I	155W	265***ri***	371\	466***9***		
071J	156X	271***dl***				
072K	162Y					
073L	165Z					
074M	172*					
	174#					

This would be easy to integrate into a multipurpose app to connect on digital modes for a low bandwidth 300 baud signal at 23 bits per character. This would be quite reliable as a means of doing a more modern RTTY. Just leaves ` _ | and ~ plus some brackets in base ASCII to do later, with 20 (11-9) codes "free". The 2xx and the 6xx lines for meta use. This gives the printable 63, and the 20 control characters with no print, along with a special control for inclusion in printing (***dl*** for delete correction) for 83.

So the 2xx codes (non-destructive locators except "delete" the anti-time locator) are colour saturation and direction control with delete (which correction "time" dynamics perhaps in a 6-bit code), and the 6xx codes are where more complex things happen. A basis repetition rate for distance starts and the coding uses this as a basis to transmit on. So a basis of 16 repetitions means each symbol is sent 16 times, for a 1/16 data rate. 612 uses 2^n repetitions based on a log for the number of rp after the symbol to be repeated. 2, 4, 8, 16 ... after rp, rp rp,rp rp rp ... 662 returns to the maximum basis of repetitions and attempts to reduce to keep the number of 627 messages down.

The basis and the use of 612 might lead to a 662 if the decoder is not in synchronization with respect to the basis of repeats. This basis repetition is ignored on the higher-level code and is just a summation of noise to increase S/N by the symbol repetition.

606 ***sy*** - synchronous idle
612 ***rp*** - repetition of x[rp]x or x[rp]x[rp]xx or x[rp]x[rp]xx[rp]xxxx etc.
624 ***ra -*** repetition acknowledge all reps in RX in TX in [ra]x format
627 ***re*** - repetition acknowledge with err correct as 624
631 ***ri*** - repetition basis increase request (2*)
632 ***rd*** - repetition basis decrease request (2/)
654 ***ok*** - accept basis repetition count by request (accepting a 631 or 632)
662 ***un*** - unsync of repetition error reply (restore maximum basis)
664 ***cq*** - followed by callsign (station ID) and ***sy*** termination

This allows for a variable data distance at a constant rate especially if the RX has a sampling of code expectation and averaging over the number of symbol reps. It also synchronizes the start of many DCS codes but would reduce the speed of lock to need the code aligned.

Extended codes could be used to extend the coding to include other things. This is not necessary, and 83 symbols are enough. This is a good start, and extras are fine though. Even precise data rate coding locking would give better performance over DX at high repetition basis.

A modified form of base64 encoding along with digital signatures (El Gamal?) could provide good binary 8-bit transmission, and block reception good certainty using the 63 non control symbols and ***sy*** for making up any block of 5 characters. A return of the good signature or the false signature on error makes for a good block retransmit given a simplex window size of 1. In this case, synchronous idle would be a suitable preamble, and the 2xx and 6xx (except 606) codes would be ignored for inclusion as part of the base64-esque stream, and would just work as control interleaved with the mapped 64 (63 plus ***sy***) symbol output. As below in column reading order (00-63) picked as sequential groups of 4 ASCII (or UTF-8) become 5 DCS characters. 606 was chosen as idle time is not a feature of blocked transmission.

023@	054H	116P	156X	331)	412***1***	466***9***	712!
025A	065I	125Q	162Y	343+	413***2***	503:	723"
026B	071J	131R	165Z	346,	423***3***	506;	731£
031C	072K	132S	172*	351-	431***4***	516<	732$
032D	073L	134T	174#	364.	432***5***	532=	734%
043E	074M	143U	306***lf***	365/	445***6***	546>	743^
047F	114N	152V	311'	371\	464***7***	565?	754&
051G	115O	155W	315(	411***0***	465***8***	703***sp***	606 ***sy***

A 612 ***rp*** can repeat itself, for triple pattern rp, rp, rp of three repeats? No this would make decode essentially the wrong type of grammar for simplicity. This provides the ***rp, rp*** double symbol to enter block mode, and standard signature digest immediately followed by the block, so that the digest can be used to obtain block termination. After each 5 DCS, the signature is checked, and match provides a decode, with ASCII NUL used as an end pack to a character multiple of 4. Further blocks need the rp double before sending, as this makes entering block mode and exit block mode easy. The rp double can also exit and re-enter block mode with a block fail.

271 ***dl*** will just exit block mode. All the other 2xx codes will just maybe animate the reception process for some matrix action with the swirly timer. This completes the standard version 1 DCS-TLBC.