

# HDCW V2.1 User Manual

Klaus von der Heide, DJ5HG

## 1. General Aspects

### 1.1. General Application Area

HDCW was designed as an RTTY-type mode for use on radio channels with great frequency spreading as in aurora. If the frequency spreading is as low such that PSK31 decodes then these coherent digital modes outperform HDCW. But also on none-spreading channels HDCW may be very useful by two reasons: (1) it is very robust against QRM, and (2) it decodes all incoming signals within the usual SSB-bandwidth in parallel.

HDCW is not a pure digital mode, because there is no final decision in the receiver, which character is to be displayed. All characters are displayed at all places in time. But their visibility is an analog function of their confidence. Therefore, only those with acceptable confidence are visible at all. It remains the operators's decision, what has been received. Pure noise produces a noisy text (see figures 3-7). As a consequence, the displayed output can only be stored as an image, not as a text.

### 1.2. Modulation

HDCW uses optionally ASK or FSK. Since HDCW receives the audio signal from an SSB transceiver, the bandwidth is limited to about 2.5 kHz. Therefore FSK is not useful if the frequency spreading is as large as 2 kHz or near to it. The spreading of aurora depends on several parameters, especially on the frequency band and on the beamwidth. On 144 MHz, FSK with a shift of 1 kHz should work well. In respect to total transmitted energy, ASK is slightly better. If the PA is peak-power limited and not mean-power limited, then FSK can transmit more energy per transmission. In that case FSK yields the better SNR.

Of course, FSK is the better choice in respect to Electromagnetic Compatibility (EMC).

### 1.3. Code

Spreading radio channels as aurora, FAI, or RS usually change rapidly. A rigid coding scheme as it is used in JT65 is not appropriate in that case. The information entity which forms a codeword must be considerably smaller, i.e. three, two, or one characters. HDCW uses the characters of the following alphabet as the information entity:

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789/.,-?@

Such a small entity enables fast dialogs as with CW or RTTY. The drawback of a small information entity is a (theoretical) loss of 3.5 dB in respect to minimum usable SNR compared to 50 bits per information entity [1,2]. An additional loss (and source of errors in operator synchronization) is caused by the necessity to communicate the changeover via the radio channel.

character	HDCW Code	Morse Code
A	1010111010011101000100000101011000001111101	10111000
B	0111000000011111110111110000001100101101000	111010101000
C	1010011101100110100101010111101000101000010	11101011101000
D	0100011011111010011101011010001001000110001	1110101000
E	0011101001000001010000111110001011101110110	1000
F	0010000011101110000110111101100011010101100	1010111000
G	1110000001001001100111000010111011110010101	111011101000
H	1001011000110100000110111000011101110000111	1010101000
I	1001000111110000010011010001111110001110100	101000
J	1110001101010011001010110101000101011010001	1011101110111000
K	1101100101011110001100100010110001001100110	111010111000
L	0010100010010010001011001011101101101001111	101110101000
M	1101101110100011100110011010000010001001101	1110111000
N	1001010001010111101010001011001010010111010	11101000
O	1110110010010100110001111011010011011000000	11101110111000
P	0100011001010011010101001100110110011001110	10111011101000
Q	0000001110001110110010000110011111011100011	1110111010111000
R	1011010110001011001001011110010100110100100	1011101000
S	0010101101111101011110001011010110100000000	10101000
T	110010010010010000110100110000111111111000	111000
U	0010101100111010101001110000011010010011110	1010111000
V	0000111011000000101110110010110100111111000	101010111000
W	1010110101100011110000101000111101000101001	101110111000
X	0011010100010100111100010110100111000011101	101010111000
Y	000111010100100011111101101001000011000101	1110101110111000
Z	1101111000001010011010010101111011100001000	11101110101000
	1100100001111000110000010111000100110101111	00
0	0111010111111000100010001100010001101011010	1110111011101110111000
1	100001001010101011111100100001000111101010110	10111011101110111000
2	0100110110110111010010100110101000110010100	101011101110111000
3	1000000000101101011011111110110000001011011	1010101110111000
4	1001101011011110100001101100100110100010001	10101010111000
5	0001011100111001000001100011100011111101001	101010101000
6	0111001000100110111000001001110000111110101	11101010101000
7	0111110000100010000111100111010110000110011	1110111010101000
8	1111001110001000010100101011101100010011010	111011101110101000
9	0100000110010001101100111101111010100100011	11101110111011101000
/	0111110011101101001000010000101110011000011	1110101011101000
.	0001100111000111010101010001010001110011011	10111010111010111000
,	0101001011100101101001100111011101000001100	1110111010101110111000
-	0100111100001101100011011001100101000110110	111010101010111000
?	1110011111000100011011100000000010100101111	101011101110101000
@	1011101010110001111111000100100001010100010	10111011101011101000

**Figure 1.** HDCW uses an alphabet of 43 characters. Each character is encoded by a binary codeword of 43 bits as shown in this figure. Each codeword has 21 ones and 22 zeros. All 903 possible pairs of different codewords differ exactly in 22 bitpositions. If a received binary word of 43 bits is decoded by search for the codeword which differs in the lowest number of bit positions then a decoding error occurs when the received word has at least 11 faulty bits, otherwise the correct character will be decoded. In contrast, the Morse code has different length and no error correcting capability at all (see figure 2)

Figure 1 specifies the complete HDCW-code and compares it to the Morse code. All HDCW-codewords mutually differ in 22 bit positions. So a false decode only occurs when more than 10 bits are inverted. In contrast, single-bit errors in CW can corrupt the information. This is demonstrated at the example of the author's callsign in figure 2.

```

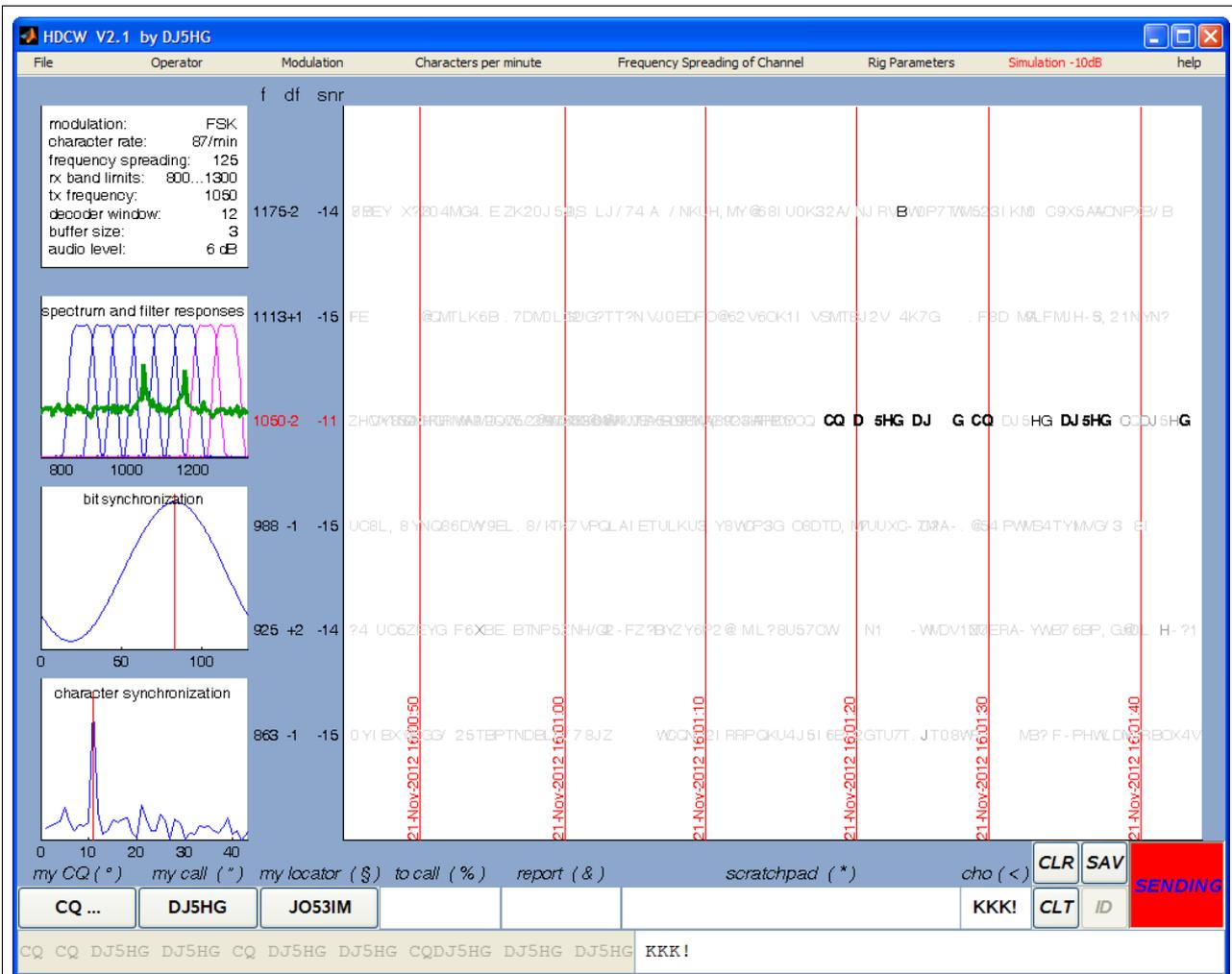
DJ5HG 111010100010111011101110001010101010001010101000111011101000
HJ5HG 101010100010111011101110001010101010001010101000111011101000
MJ5HG 111011100010111011101110001010101010001010101000111011101000
D35HG 111010100010101011101110001010101010001010101000111011101000
DJBHG 11101010001011101110111000110101010001010101000111011101000
DJLHG 11101010001011101110111000101101010001010101000111011101000
DJFHG 1110101000101110111011100010101101010001010101000111011101000
DJVHG 11101010001011101110111000101010110001010101000111011101000
DJ5DG 11101010001011101110111000101010101000110101000111011101000
DJ5RG 111010100010111011101110001010101010001011101000111011101000
DJ5UG 11101010001011101110111000101010101000101011000111011101000
DJ5HF 111010100010111011101110001010101010001010101000101011101000
DJ5HB 111010100010111011101110001010101010001010101000111010101000

```

**Figure 2.** In rapid fading conditions a dash of the Morse code may be read as two dots or two dots as a dash which in most cases cannot be detected as an error. This figure shows the author's callsign in Morse code (upper line) and the twelve single-bit errors of this type. In contrast, the HDCW code can correct up to 10 faulty bits in each binary word (at the price of nearly three times the number of bits).

#### 1.4. Transmission Speed

HDCW implements seven different transmission speeds between 349 and 5.5 characters per minute. The precise values are:  $60 * 8000 / 43 / 2^k$  with  $k = 5 \dots 12$ . The value 87 characters per minute may be the most adequate for normal QSOs. It is similar to usual CW speed. The corresponding bitrates are  $8000 / 2^k$  with  $k = 5 \dots 12$ . The bitlength is  $2^k$  samples (at samplerate 8000).



**Figure 3.** HDCW GUI. The two lines at the bottom are for transmitter input and control. All above these lines is receiver output. Only the menu at the top is for both, receiver and transmitter control.

- (a) The upper left field shows the most important actual parameters. The fields below show:
- (b) The actual spectrum and the bandpass responses of the running receivers.
- (c) The quality of the bit synchronization (which is perfect here).
- (d) The quality of the character synchronization (which also is perfect here).

The large field to the right shows the decoder output (one line per running receiver).

The corresponding center frequency (f) is given in the vertical bar with actual frequency offset (df) and SNR. df of the 1050 Hz receiver says that the received signal probably is about 2 Hz lower, which is below the accuracy of frequency detection.

The 1050 Hz receiver is colored red. It was selected by a mouseclick as the primary receiver. The primary receiver has priority, while the others are computed as far as the computer's processors can do the job. The figures for bit synchronization and character synchronization are from the primary receiver.

The button CLR clears the decoded text, the button SAV saves the HDCW-window as a JPG-image. The large green button is for change-over.

The lowest line is a text-input line for text to be sent. Some buttons CQ..., station call, and station locator are for pasting this information into the text line. This also is possible with free text written into the edit fields for "to call", "report", "scratchpad", and "changeover".

The button CLT clears the lowest line. The button ID adds the station call in CW.

## 2. User Interface

The Graphical User Interface (GUI) of the HDCW-program is shown in figure 3. The two lines at the bottom are controls for the transmitter. Everything above belongs to the receiver. Only the menu bar is for control of both, receiver and transmitter.

### 2.1. Menu Bar

The menu bar has 8 items:

File    a wave-file can be selected as input to the receivers instead of the sound

Operator            input of station callsign, Maidenhead locator and preferred CQ call

Modulation        choose ASK or FSK

Character per minute    choose the character rate: 349, 174, 87, 44, 22, 11, 5.5 characters / minute

Frequency spreading    choose appropriate value between 500 and 2500 Hz or no spreading

Rig Parameters        set several important parameters as tx frequency, bandlimits of receiver, etc.

Simulation         set simulation on/off or set channel type and SNR

Help                 display short instructions on HDCW

### 2.2. Receiver Output and Transmitter Input

This is explained in figure 3. A detailed description follows in Chapter 4.

### 2.3. Graphic Renderer

On some systems, the OpenGL renderer does not correctly display the generated graphic. HDCW starts with the zbuffer renderer by that reason. But the OpenGL renderer usually is better and faster. You can toggle between both renderers by a double click on the gray background surrounding the graphic text field.

## 3. The Decoding Process

There is an important difference between the HDCW-decoding and digital receivers as PSK31, RTTY, JT44, or FSK441. The HDCW receiver converts the incoming noisy signal into an outgoing *noisy* text stream. As a consequence, the final decision stage is the operator himself as in reading CW. Figure 3 gives an impression. The confidence of the letters is gray-tone coded. So most of the output is more or less gray colored garbage. But typical noise also produces isolated "significant" output. While this may be very irritating in JT65 with its complete and senseful message templates, the isolated bold letters in HDCW are easily identified as noise-peaks. Of course, the gray color could simply be changed into white. But the author feels that the visibility of the noise offers more insight into the process of information transfer. A drawback of this analog output is the difficulty to store the decoded information in a usual textfile. At the moment, only the option to store the whole GUI as a \*.JPG image is implemented (SAV-button).

## 4. The Menu Bar of HDCW

### 4.1 File

Besides the "exit HDCW" this menu item offers three options:

#### 4.1.1 Set Record Folder

HDCW uses a folder with a fixed path for records. If you want to generally store the records and screenshots somewhere on your computer then specify the folder here.

#### 4.1.2 Save as PDF

This offers the possibility to store the HDCW-GUI as a \*.PDF file.

#### 4.1.3 Decode Sound Input

This is the normal operation mode with the input from the soundcard. HDCW starts in this mode.

#### 4.1.4 Decode Record

Select this item to specify a wave-file which then will be decoded with the actual receiver settings. If you are not sure, set the bandlimits to 500 ... 2500 Hz, the modulation to ASK, and try all speeds. All standard samplerates (8000, 11025, 16000, 22050, 32000, 44100, 48000) are supported.

### 4.2 Set Local Parameters

#### 4.2.1 Operator

Select the menu option "**Operator**" and the submenu item "**set call & locator**". A small window will pop up. You should place this window somewhere outside the main window of HDCW. Then enter your call and locator and quit by clicking the ok button.

Do the same with the submenu item "**preferred CQ call**" within the menu option "**Operator**". This entry should contain the complete transmission including possible repetitions, the changeover, and the final "!"-character as a command for changeover.

### 4.3 Set Operating Parameters

#### 4.3.1 Modulation

Select the modulation **ASK** or **FSK** with the menu option "**modulation**". FSK uses a fixed shift of 1000 Hz for spreading channels or four times the bitrate (see 1.4) for none-spreading channels.

#### 4.3.2 Characters per Minute

The transmission rate is selected with the menu option "**Characters per minute**". 87 is a good value for your first tests since it is similar to the usual CW speed.

### 4.3.3 Frequency Spreading of Channel

The menu item "**Frequency Spreading of Channel**" offers 5 options ranging between 500 Hz and 2500 Hz". If FSK is used then spreading of 1500 and 2500 Hz is disabled. A further option is "**no spreading**". In that case the receiver uses a narrow-band filter of four times the bitrate. In this case, the FSK-shift is 1000, 500, 250, 125, 62.5, 31.25, 16.125 Hz for the character rates 349, 174, 87, 44, 22, 11, 5.5 per minute.

## 4.4 Set Rig Parameters

Parameters which are set only once or are rarely changed are collected under the menu item **Set Rig Parameters**. These are:

### 4.4.1 Tx Audio Frequency

Set the audio carrier frequency which is the input to the SSB-transmitter. A warning will be given if this frequency is outside the specified receiver band limits. But working in split-mode may be very useful in pile-ups or in aurora with it's huge Doppler-shift. In FSK-mode tx frequency (and rx-frequency too) generally is the lower audio carrier. Make sure to use USB with your transceiver.

### 4.4.2 Rx Audio Band Limits

Set the two frequencies for lower and upper band limits in Hz, both within 500 Hz and 2500 Hz. You could leave these values at their default values 500 Hz and 2500 Hz. But in "no spreading" mode this can lead to a large number of receivers which may overcharge the computing capability of your PC.

### 4.4.3 Delta F

The computation of the frequency shift (the df value) needs considerable computing power. If you are not interested in it (at large spreading for example) you should set Delta F to OFF.

### 4.4.4 Decoder Window

The decoder window specifies the length of the input signal. If you choose  $m$  characters the receiver will analyse the last  $n$  incoming samples, where  $n = m * 43 * 2^k$  with  $k = 5 \dots 12$  as in Chapter 1.4 . Especially the bit synchronization and the character synchronization is based on the sequence of these  $n$  samples. If the received message is considerably shorter than  $m$  characters then noise from outside the message can prevent synchronization. If on the other hand a very weak message is longer than  $m$  characters then the actual synchronization does not utilize all energy of the complete message and may fail by that reason. A window of 12 characters probably is a good compromise for most purposes.

### 4.4.5 Buffer Size

The output signal is generated and sent to the sound card character-by-character. The advantage is that the text to be transmitted can be written and even corrected while the transmitter is running. The drawback is the necessity of a buffer between the sound card and the send text of the GUI, even worse: the drawback is that this buffer can run empty while your computer is busy doing windows-updates, anti-virus checks or something else other than putting the next character into the HDCW-output buffer. The HDCW program does not abort if the buffer runs empty. It tries to evaluate the

number of samples the soundcard output was interrupted, throws away these samples, fills the buffer with samples to be sent from the actual moment (or better: what the program believes to be that moment), and starts the soundcard again. But the author cannot guarantee that this procedure never goes into nirvana after itself again was interrupted several times by windows.

To avoid such problems choose an appropriate value for the buffer size on your computer. For normal QSO mode and character rates of 87 or lower the buffer size 2 should be sufficient. At higher speeds use 3 or 5.

Real buffer problems may occur in simulation mode because the receivers generate a heavy workload in parallel to the transmitter. The delayed transmission caused by a large buffer size is not a problem in a simulation.

#### **4.4.6 COM Port**

Here you can select a COM-port for transceiver control.

#### **4.4.7 PTT**

Options for the PTT are: RTS; RTS inverted; DTR; DTR inverted.

#### **4.4.8 Sound Input ID**

HDCW detects all activated sound input devices of your computer. By default the device ID 0 is chosen. If you want to use a different device, select it here. You must quit then HDCW and start it again to activate the change.

#### **4.4.9 Sound Output ID**

Similar to the sound input also the device ID for the sound output can be chosen. Different IDs for sound input and sound output are allowed. If you change the ID, you must quit HDCW then and start it again. The output ID may be different from the input ID of the same device. Sorry, names of the output devices are not available.

### **4.5 Simulation**

HDCW has an inbuilt simulation capability which is controlled via the menu item **simulation** with 3 submenus:

#### **4.5.1 ON / OFF**

Here you can toggle between simulation ON and simulation OFF. If the simulation is on the transceiver control via the COM-port is disabled.

#### **4.5.2 Channel**

Options of this submenu are "**no fading**", "**fading 1s**", "**fading 0.2s**", "**aurora**".

#### **4.5.3 SNR**

Here you can set the SNR value for the simulation.

## 5. Details of the HDCW-GUI

### 5.1 General Features of the GUI

The HDCW-GUI has no fixed format. The fontsize and the fontpitch automatically are adapted to the actual size of the GUI. The user should change the GUI-size to optimize the readability of the decoded output in respect of his personal needs.

### 5.2 Interpretation of the Receiver Output

#### 5.2.1 The Spectrum Display

The spectrum of the received signal is colored green as in figures 3 and 4. The filter responses of the receivers are colored blue (ASK and lower frequency in FSK) and magenta (upper frequency in FSK). If the frequency spacing of the receiver coincides with the frequency shift as in figure 3 then the blue lines overwrite the magenta lines. Therefore only the upper frequency responses of the two receivers at the higher band limit are visible.

A simple mouse click into the spectrum display toggles between the full scale of 0 ... 3000 Hz and the close view restricted to the interval defined by the band limits.

#### 5.2.2 The Bit Synchronization

Let the decoder window be  $m$  characters. Then the signal to analyse is  $m * 43$  bits long. Since the length of a bit is  $2^k$  samples ( $k=5...12$ ) (see Chapter 1.4) the total length of the signal is  $m * 43 * 2^k$  samples. After filtering and demodulation the signal is cut into  $m * 43$  pieces of  $2^k$  samples. All these pieces are added like vectors. After proper vertical scaling the result is displayed as the bit synchronization.

A good signal looks similar to a sine wave as in figure 3, a weak signal or pure noise leads to a noisy display as in figure 4.

A sine wave is fitted into the display. The position  $s$  of its maximum is marked by a vertical red line. The received bits are read as real values from the demodulated signal at the indices

$s + (i-1)*2^k$  with  $i=1...m*43$ .

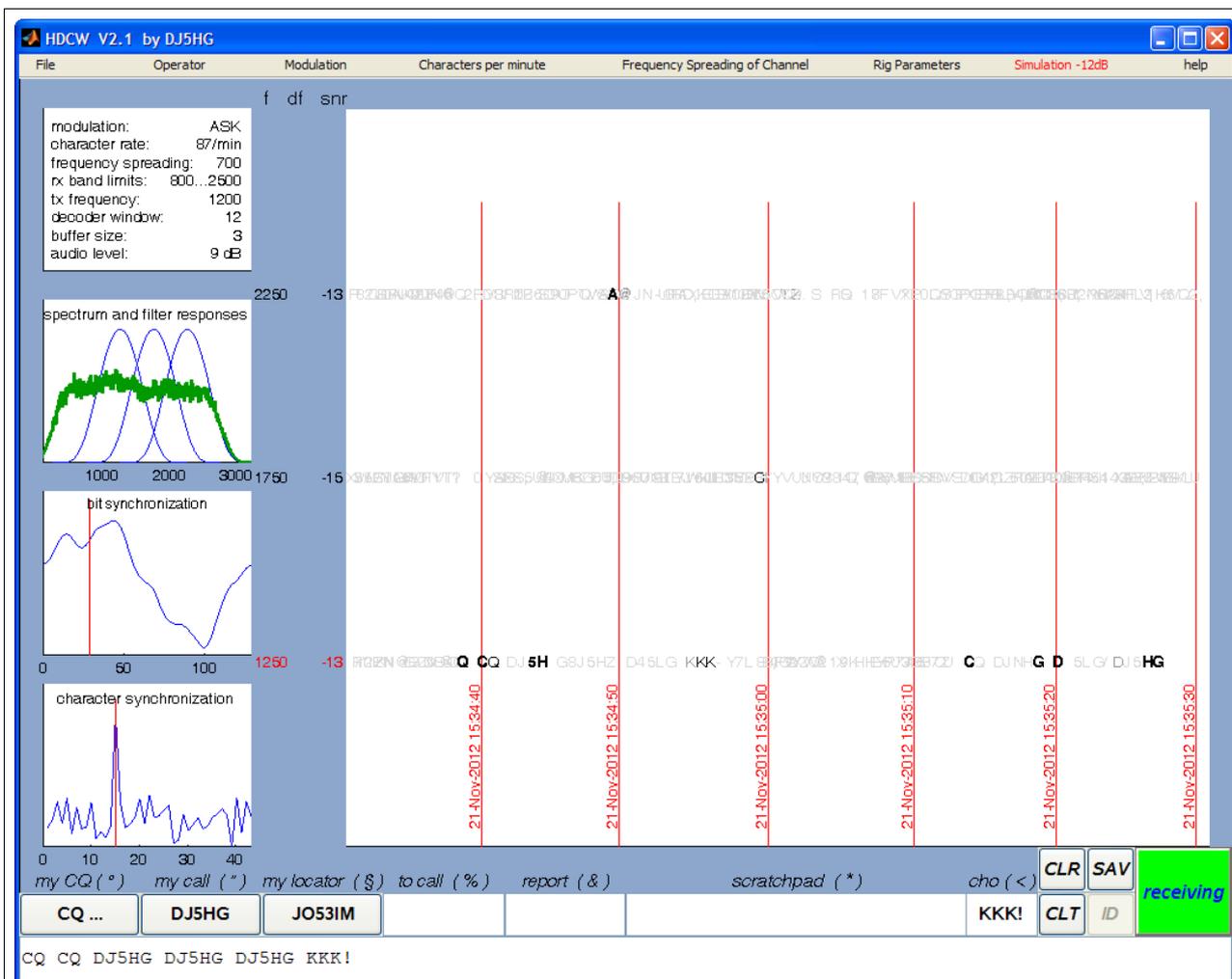
#### 5.2.3 The Character Synchronization

The principle of character synchronization is the same as that for HD43 which is described in [3]. A significant synchronization is characterized by a single peak out of 42 noise values. Figure 5 shows a case of pure noise.

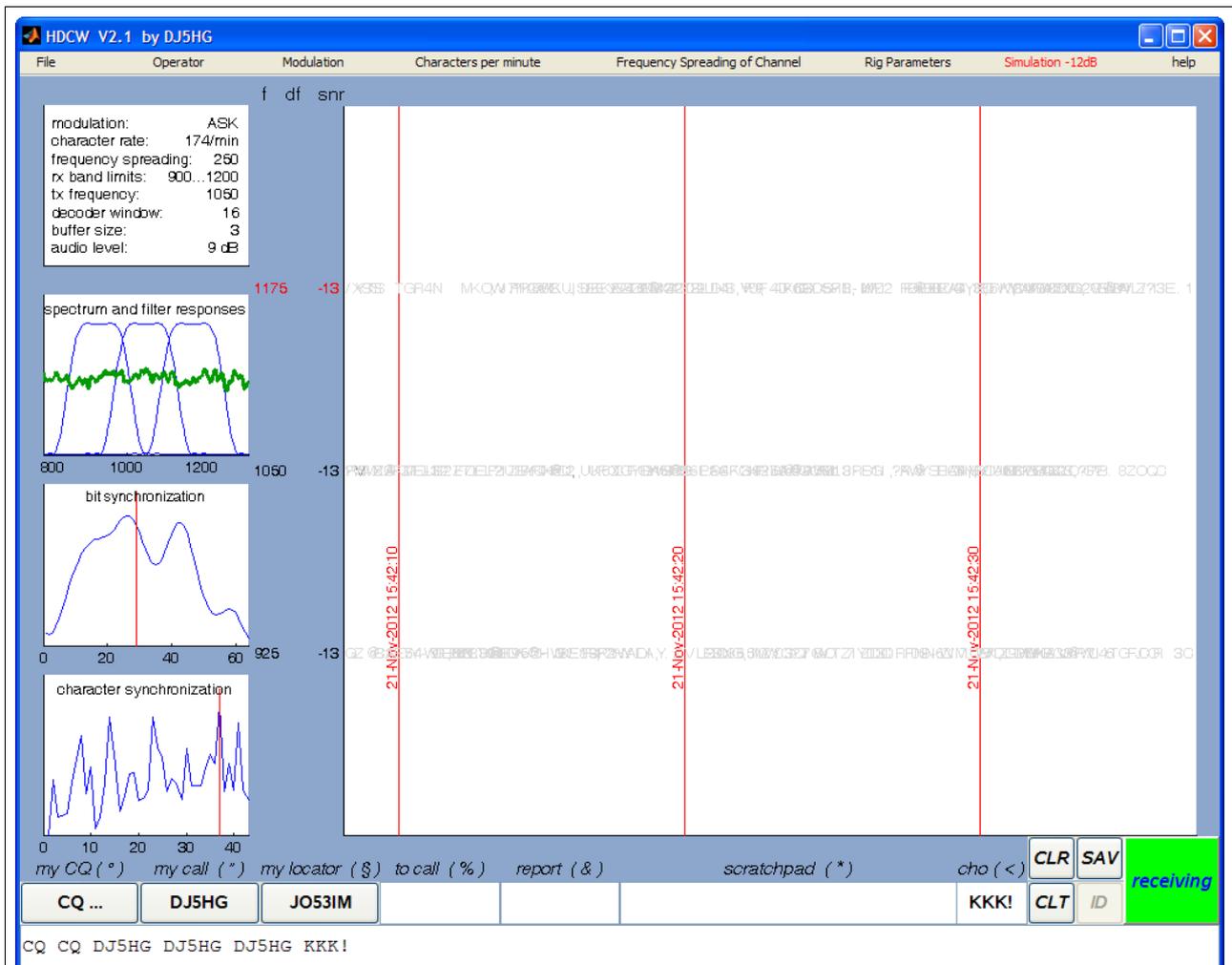
#### 5.2.4 The Decoded Text Display

The primary receiver decodes the actual window at every incoming character. As a consequence all characters will be decoded  $m$  times if  $m$  is the length of the decoder window. This is described in figure 6.

The horizontal axis of the display is time. The actual time is inserted by vertical red lines at appropriate spacing. The vertical axis is frequency  $f$ .



**Figure 4.** Example of a simulation with a weak aurora signal in ASK at -12 dB. The green spectrum display indicates the weak signal around 1000 Hz. Three receivers are running with overlapping filters separated by 500 Hz. The bit synchronization is not good, but the character synchronization shows a significant peak. The receiver sees a signal of -13 dB.



**Figure 5.** The spectrum range is restricted to the input range here. The character synchronization does not show any significant peak. Indeed, this is a result of pure noise (nothing sent in the simulation).

Let the transmitter send the text 'TEST SIGNAL OF HDCW' and let the decoder window be  $m=8$  characters. When the receiver has gotten half the signal then the actual signal contains 'TEST SIG'. The receiver evaluates the bit synchronization and finally decodes the sequence of letters and displays it. One letter later the receiver again is invoked. Now it gets 'EST SIGN', the next time 'ST SIGNA', and so on. Each time, the decoded text is displayed. The earlier is not erased.

The decoded letters are written on the decoded text display exactly where they are seen by the receiver, depending on the bit synchronization and on the character synchronization. Each character is decoded  $m$  times. So it is printed  $m$  times nearly at the same horizontal place if the character synchronization is the same. The position then only varies because of possible differences in the *bit synchronization*.

If the *character synchronizations* of the  $m$  overlapping decodes differ, then different characters are printed at shifted positions. This leads to the typical garbage of HDCW as seen in the decoded text below (left and right from the decoded 'TEST SIGNAL OF HDCW':

**Figure 6.** The nature of garbage and decoded text in the HDCW display

### 5.2.5 Consequences of Low Processor Power

Any PC should be sufficient for QSOs in HDCW. But differences will be evident in simulation mode and especially in monitoring the whole 2 kHz Bandwidth in "no spreading" mode. Figure 7 gives an example of unacceptable overload by running 63 receivers with DF switched ON. If you see these white space bands diagonally running over the decoded text display then first switch Delta f to OFF.

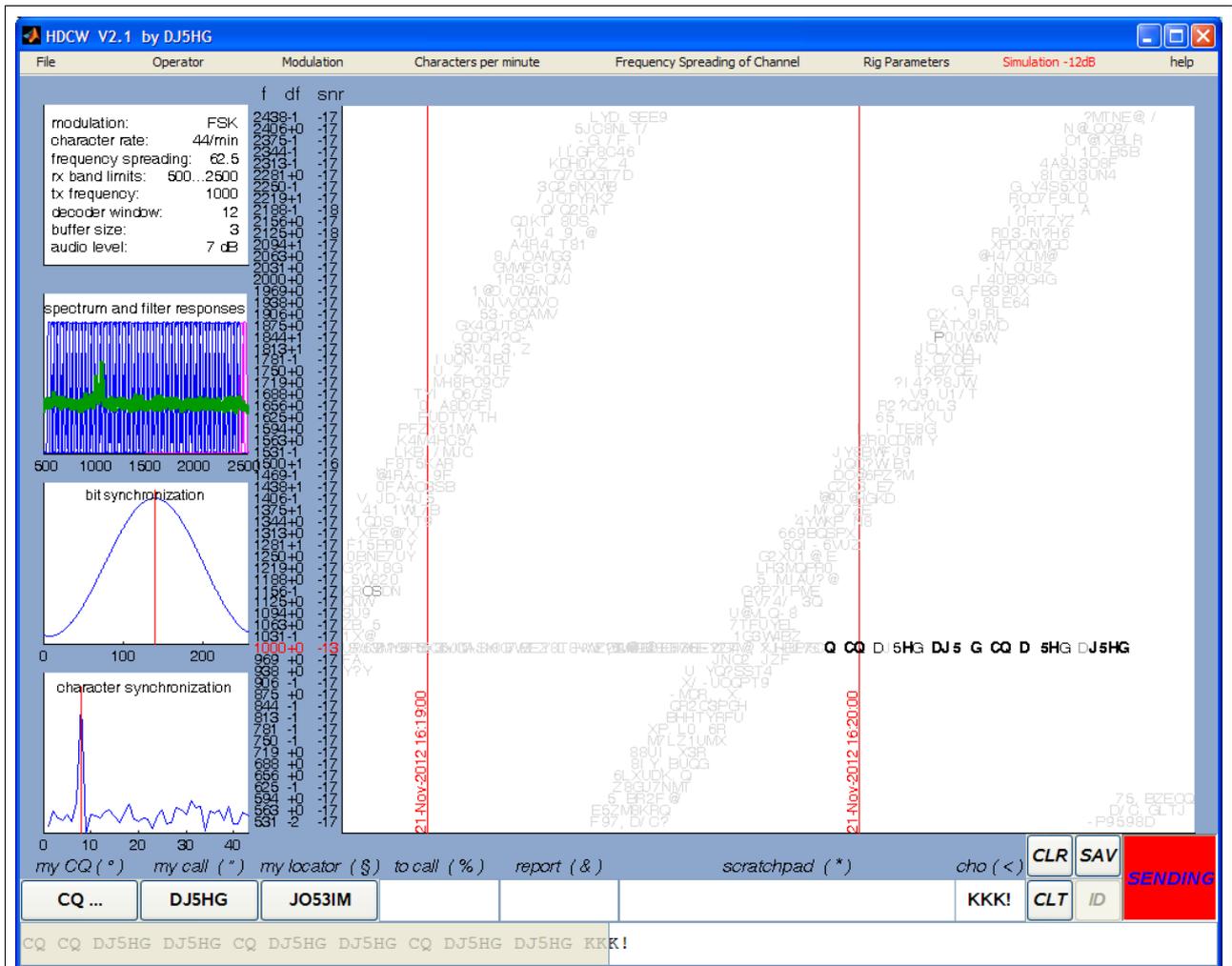
Then you have to decide between the options:

- (1) Is it better to monitor a large bandwidth with these visible gaps (may be sufficient for monitoring)
- (2) or is it better to monitor a small bandwidth without gaps (or something between)

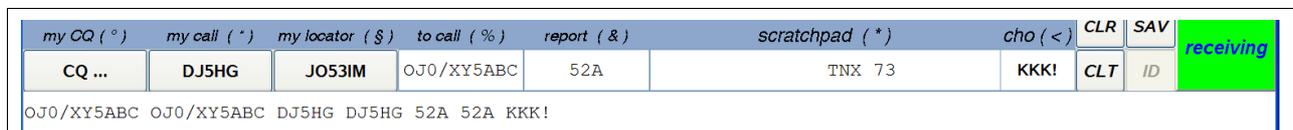
Choose the band limits correspondingly.

The receivers are invoked one after the other. In the case of figure 7, running through all 63 receivers lasts about 50 seconds. As a consequence, each of the receivers can decode only one window of (in this case) 16 characters every 50 seconds.

In contrast, the selected primary receiver runs permanently as is obvious in figure 7. So if this situation happens in a QSO that does not affect the contact if the correct frequency was selected as the primary frequency.



**Figure 7.** Effect of overloading the processor by too many receivers. Note that the selected primary receiver is running permanently (44 characters per minute, no spreading, full bandwidth) while the others get the processor at so large timesteps that they only can decode about 15 % of the signal.



**Figure 8.** To compose the sendtext 'OJ0/XY5ABC OJ0/XY5ABC DJ5HG DJ5HG 52A 52A KKK!' the call and report were entered into the corresponding edit fields. Then the text 'to call (%)' was clicked two times, then 'my call (")' two times, then 'report (&)' two times and 'cho (<)' once.

## 5.3 The Transmitter

### 5.3.1 The Change-Over Button

Whenever the green button 'receiving' is pushed it will change into red color and transmit the text content of the bottom line of the HDCW-GUI in the mode specified by the actual parameters (see figure 4). There is no effect if the send text field only contains blanks or nothing.

### 5.3.2 Composing a Message

The send text can simply be entered by setting the cursor into the line and then typing on the keyboard.

There are seven additional text fields. The three fields for '*my CQ*', '*my Call*', and '*my locator*' are predefined by the menu and cannot be edited directly. Instead, they are pushbuttons. Any mouseclick on such a button appends the corresponding text at the end of the send text line. Alternatively, you can click on the text above the buttons: '*my CQ ( " )*' etc..

The fields '*to call ( % )*' etc. can be edited. Therefore these are not buttons. The contents only can be appended to the end of the send text line by mouse click on the text '*to call ( % )*' etc.

Figure 8 gives an example of text composition.

The special symbols ° " § % & \* < do not belong to the HDCW alphabet. If they are used in one of the text edit fields they are replaced by the contents of the corresponding text field. The same send text of the example in figure 7 would result by typing into the send text field the shortcuts: %%"&&<

There are two additional special symbols:

- ! This forces the HDCW transmitter to change over to reception.  
Everything right from a ! will never be sent.
- > This symbol forces the HDCW transmitter to repeat the send text to the left of it until the transmission is stopped manually via the receiving/sending toggle button.  
Everything right from a > will never be sent. This symbol does not work in simulation mode.

Leading and trailing blanks of the send text are deleted before changing over to transmit. If there are only blanks in the send text field, the transmitter will not switch on.

### 5.3.3 Progress of Transmission

If the transmitter is running it puts the generated signal character by character into the buffer between the HDCW program and the sound card. A grey bar over the send text field indicates the progress which characters of the send text already went into the buffer. In the example of figure 4 the CQ call completely went through the buffer with the exception of the last 'K'. Since the buffer size in that example is 2 the buffer only contains the preceding 'KK' and the CQ call already was sent.

When the transmitter reaches the end of the send text but does not find a change-over symbol (!) then the transmitter transmits a blank. If the operator does not append further send text in time HDCW will automatically change-over after three added blanks in sequence.

### 5.3.4 Editing while Transmitting

The send text to the right of the grey progress bar can still be edited while the transmitter is on. But editing near to the progress bar is not recommended. Time is inexorable and probably faster than you expect. This may lead to corrupted text at the other end.

### **5.3.5 Clearing the Send Text**

The send text is cleared by pushing the CLT button.

### **5.3.6 CW Identification**

If the ID-button is pushed before a change-over to transmit then the station call will be sent in normal CW on the specified tx-frequency directly before the next HDCW-transmission. If the ID-button is pushed while a HDCW-transmission is running, the CW-identification will be appended at the end of the transmission. The CW identification is disabled in simulation mode.

## **6. References**

- [1] K. von der Heide, DJ5HG: Possibilities and Limitations of Forward Error Correction in Minimal QSOs, Proceedinds of the EME Conference Würzburg, 2006.
- [2] K. von der Heide, DJ5HG: Digital Weak Signal Communication for Minimal QSOs, DUBUS vol. 38, 1/2009, pp.36-46.
- [3] K. von der Heide, DJ5HG: HD43: a new digital mode for very weak signals, DUBUS vol. 38, 4/2009, pp. 47-57.