

CW Mode for the NUE-PSK Digital Modem

By Dave Collins, AD7JT, and George Heron, N2APB

Operating PSK-31 without a PC has been a big capability of the NUE-PSK Digital Modem for several years. RTTY support was added along the way and even more advanced digital modes continue to be in development. But modem owners are also clamoring for support of the oldest digital mode of them all ... CW!

This paper describes our open source implementation of this venerable digital mode as an implementation of the Goertzel DSP filtering technique used on the modem's dsPIC33F processor. Innovative approaches are used to solve the classic problems of speed tracking (up to 50 wpm!), adaptive weighting adjustment, and tuning.

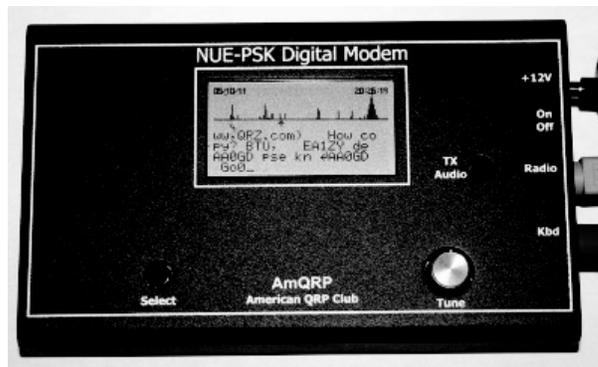


Figure 1: NUE-PSK Digital Modem. Just plug in 12V power, standard keyboard, and cable to your SSB transceiver in order to operate the digital modes ... now including CW! 2 kHz live spectrum displays band activity. User moves cursor to desired signal and decoded receive text displays in bottom four lines. Keyboard-entered characters display during transmit. Rx and Tx text saved to USB stick plugged into side of modem. Time and date shown at top of display.

INTRODUCTION

The NUE-PSK Digital Modem project was introduced to the amateur radio community some four years ago at the TAPR DCC conference. Designers W8NUE and N2APB then demonstrated the first ever *fully embedded* implementation of a PSK31 station – one that does not require a PC and soundcard to encode/decode standard PSK31 and other digital mode tones. Operators quickly grew to love the ease of just connecting a common keyboard and an SSB transceiver to the little black modem to use digital modes in the field, on a mountaintop or in the car.

Users also correctly assumed that by adding CW support to the modem they could take advantage of the same portability and relative accuracy of copy. That is, by generating CW using a keyboard for transmit (e.g., “Morse keyboard”) and decoding Morse characters to the modem’s LCD display (e.g., Morse reader) for receive, they could continue enjoying their beloved CW in a totally unique way.

So this is where AD7JT joined design team to work with N2APB in order to add this “new” digital mode (CW) to the

NUE-PSK modem platform. The transmit side of providing CW support is relatively straightforward and pretty easily accomplished. Morse keyboards have been around for some time – actually since the early years of RTTY – and a simple look up table (LUT) can be implemented to turn an ASCII keystroke into its corresponding Morse code DIT and DAH sequence.

However the receive side has classically been more difficult to implement well. Traditional Morse readers in the market typically use selective audio filtering with analog components to isolate the CW tones coming from the radio’s speaker. While some may work moderately well when the signals are strong and the band is clear, most have problems under adverse band conditions of QSB, QRM, QRN and QLF (i.e., when the operator is sending with his left foot!)

In fact the quality of the CW being sent by the other side usually is the biggest factor of an automated decoding system’s ability to display the intended message. Most CW ops use automatic keyers these days, thus somewhat standardizing the transmitted code; but even so, the weighting of the DITs and DAHs, and the inter-char/word

spacing often poses a grave challenge to most Morse readers.

Based on the ideas presented at the outstanding “DSP Short Course” delivered by KN6KB at last year’s DCC, the authors decided to tackle these problems using a Goertzel-based audio filter implemented on the NUE-PSK’s internal dsPIC processor. This DSP function sports fast operation, is able to resolve narrow bandwidths, and was relatively straightforward to program in C code. Further, the FFT software framework was already in place on the modem’s spectrum display, so dropping the code in should be quite easy, right?

Well, we all know how these “easy” things go, but that was the nature of the project being presented here! In addition to the DSP filtering to pick out the specific ON-OFF keying of the CW signal, we needed to add many side features, bells and whistles in order to make this a workable mode on the modem. ...

- Goertzel filter to detect the tone
- Morse-to-ASCII conversion
- Automatic speed detection & tracking
- Automatic DIT-DAH weighting adjustments
- AFC (to track drifting signals)
- Tuning algorithm & indication
- Ways to represent special CW chars, prosigns, etc
- Saving CW QSOs on the USB drive

We believe the results are quite good and collectively make for an interesting new capability for the NUE-PSK modem. Implemented solely in software, no hardware additions are needed; any of the 1,000+ owners today may update their modems with the new version containing this CW mode support using the built-in boot loader capability.

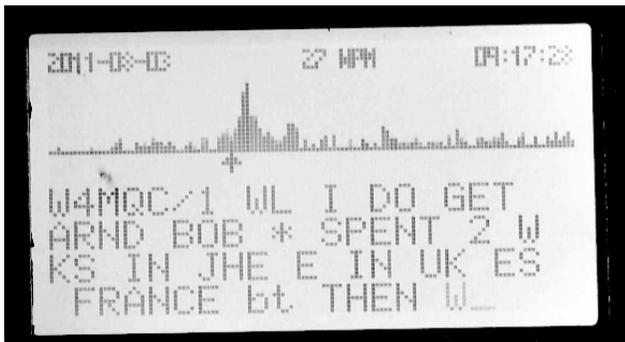


Figure 2: Photo of modem’s display, capturing part of 27 wpm CW QSO-in-progress on Aug 8, 2011 at 9:17am. Notice prosign BT and an undecipherable character (*).

Moreover, we hope that this implementation of a high-performance, embedded CW decoder / transmitter on an open software platform such as the NUE-PSK modem may give some other technologists and experimenters some ideas for extending the techniques on this dsPIC processor, or on other projects here in our ham community.

And lastly, this project provides a delicious way to reinvigorate interest for the oldest digital mode of them all ... CW!

DEFINITIONS

Before getting into the details, we’d like to provide some definitions. Much is obvious, but it will be helpful to review our common understanding of the basic elements of Morse code.

Throughout this paper the term “SPACE” refers to a key-up condition and the term “MARK” refers to a key-down condition. The term “DIT” refers to the MARK duration of a Morse code dot and the term “DAH” refers to the MARK duration of a Morse code dash. DITs and DAHs are referred to as “Morse elements”. Morse elements are combined to form characters. Characters can be letters, numbers, punctuation marks, and prosigns. A prosign is a combination of elements representing a special character and, in this context, something unique to amateur radio. Throughout this paper, prosigns are represented by two, lower-case, boldface letters the Morse codes for which are sent, run together, with no inter-character space between them. Familiar prosigns include **bk** (Break) and **sk** (Silent Key). Normal, alpha Morse characters are represented as upper-case, boldface letters.

The terms “intra-character SPACE” and “inter-element SPACE” refer to the duration of the SPACE between DITs and DAHs within a character. The term “inter-character SPACE” refers to the SPACE duration between characters. The term “inter-word SPACE” refers to the SPACE duration between words.

A basic time unit (T_{cw}) is used to quantify time durations of Morse code MARKs and SPACES. Standard Morse code time durations are as follows:

DIT = 1 x T_{cw} (“short MARK”)

DAH = 3 x T_{cw} (“long MARK”)

Inter-element SPACE = 1 x T_{cw}

(between DITs and DAHs within a character)

Inter-character SPACE = 3 x T_{cw} (between characters)

Inter-word SPACE = 7 x T_{cw} (minimum, between words)

Since all timing is derived from Tcw, its duration determines the transmission rate commonly stated in “words per minute” or “wpm”. A standard “word” has been defined for computing wpm. That word is “PARIS ” which, in Morse code, has ten DITs and four DAHs and contains exactly 50 Tcw including the ending inter-word SPACE. Using this standard word, the relationship between Tcw and wpm is: $wpm = 1200 / Tcw$, where Tcw is expressed in milliseconds and all element and SPACE durations are standard as listed above.

GENERAL DESCRIPTION

While receiving CW, the modem will copy and display standard Morse code characters received through an audio output from a transceiver. While transmitting, the modem will transmit standard Morse code characters as entered by the operator on the keyboard or contained in macros via the audio input to a transceiver. The transceiver shall be in DATA, PACKET, or SSB mode, **but not CW mode**. The CW operating mode is selected for the modem using the MODE entry in the configuration menu system. Once CW mode is activated, the modem will be in the receive state until transmission is activated.

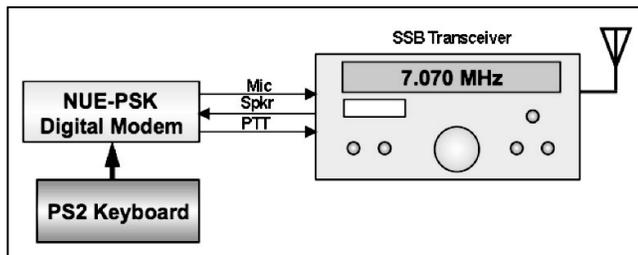


Figure 3: Modem connects to transceiver and delivers modulated audio tones to operate the digital modes ... even in mode A1 (CW). ON-OFF delivery of audio tone in SSB mode yields Morse code.

CW RECEPTION

In order to accurately receive CW, the signal must have certain characteristics often attributed to an operator with, what is commonly referred to as, a “good fist”.

Best results will be obtained by the modem when the signal is generated by a computer (or a microcontroller) and strictly follows normal Morse code standards (e.g., a DAH is three times a DIT, inter-word SPACES are at least 7 DITs, etc.). This case includes modem-to-modem CW communication. The most probable causes of errors will be received signal fading (QSB), interference from other stations operating near our operating frequency (QRM), or

excessive noise (low Signal-to-Noise ratio) in the receive channel. Some of these sources can be handled through filtering, Automatic Gain Control (AGC), or adjusting the MARK-SPACE threshold level. There are limits, however, and solid CW copy by the modem will require a fairly strong, distinct, and clear tone from the receiver audio channel with the receiver tuned to produce a CW side tone in the 1 KHz audio spectrum processed by the modem in CW mode.

The second-best choice would be an electronic keyer operating in fully automatic mode so that it will strictly control the duration of DITs and DAHs and the SPACES between them. In this mode, the transmitting operator is responsible for inter-character and inter-word SPACE duration. The receive algorithm will try to accommodate variations in this timing but there are limits. (Some of the parameters used to define various CW elements can be tweaked manually by the operator.) The following are some of the problems that will occur when these limits are exceeded:

- Inter-word SPACE too short** ... words run together
- Inter-character SPACE too long** ... extra spaces between characters
- Inter-character SPACE too short** ... characters run together

The third-best choice would be an electronic keyer operating in semi-automatic (“bug”) mode where the keyer controls the DIT duration and the SPACE between DITs but the operator is responsible for DAH duration, the inter-DAH SPACE, and those items listed above for an electronic keyer in fully automatic mode. Exceeding limits in these areas adds the following to our list of problems:

- DAH duration too short** ... DAHs interpreted as DITs
- Inter-DAH SPACE too long**chars split into two or more false chars
- Inter-element SPACE too short** missed DITs (combined into long DAHs)
- Inter-DAH SPACE too short** ... missed DAHs (combined into very long DAHs)

These conditions can also cause the modem to recalculate an inaccurate Tcw which will affect the accuracy of decoding following characters.

CW TRANSMISSION

During CW transmission, the operator selects a transmission speed (words per minute or “wpm”) and all Morse elements, inter element, and inter character durations are determined by that selection. A minimum inter-word duration is determined by the modem but this duration may

be extended by keying additional spaces between words or by pauses in the keyed input. The minimum, transmitted inter-word SPACE duration is set at seven Tcw. Transmitted inter-element SPACE durations are set at one Tcw; inter-character SPACE durations are set at three Tcw.

To avoid injecting some of the problems described above in the transmitted Morse code and to decouple the result from the operator's keying ability as much as possible; text is transmitted in whole words only. As characters are keyed, they are displayed and buffered until a space is entered. At this point, all characters up to and including the space are processed and transmitted. This assures proper spacing between elements, characters, and words in the transmitted text.

FUNCTIONAL DESCRIPTION

An SSB transceiver translates received RF to baseband audio during reception and translates base-band audio to RF during transmission. The modem demodulates the received audio signal to extract the digital information received and modulates the generated audio signal to insert the digital information to be transmitted.

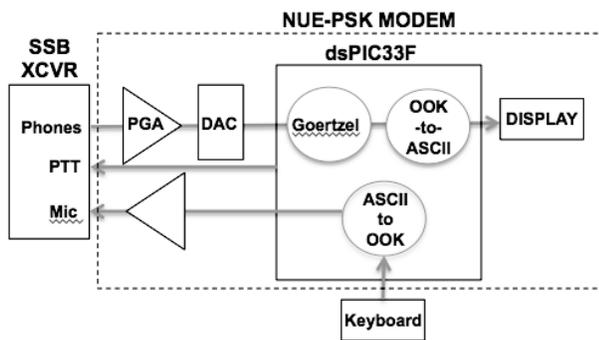


Figure 4: Signal Processing Chain. Audio tones delivered by transceiver to modem for digitizing. Goertzel filter determines energy content at specified cursor frequency. On-Off-Keying (OOK) processed to determine weighting, speed and ultimately the text character to be displayed. On Tx, keyboard presses turned to OOK and sent to transceiver when PTT active.

CW RECEPTION

Baseband audio received from the transceiver is amplified by a Programmable Gain Amplifier (MCP6S21 PGA) and digitized by one of the analog to digital channels in the modem controller. The digitized audio is sampled 8,000 times per second and processed by the firmware to extract the MARK and SPACE information and generate Morse code characters. These characters are translated to ASCII

and displayed on the lower four 20-character lines of the modem's LCD display.

Goertzel Filter

A specialized DFT algorithm called a "Goertzel filter" is used to process the raw ADC output. The Goertzel algorithm is very fast to compute and very efficient when the application only requires analysis of a small number of frequencies, when only energy levels are needed, and when phase information is not needed. The most common use for Goertzel filters is to decode the DTMF tones generated by telephone keypads.

The real beauty of the Goertzel filter algorithm is its simplicity. The code is so compact that it can be executed in the interrupt service routine 8,000 times a second when the digitized received audio is sampled. The math behind the Goertzel algorithm is quite complicated and will not be presented here. Instead, the implementation is presented. The following code snippet is executed once every 125 us to process each digital sample (f_sample) as it is taken:

```

. . . . .
g_sample = (((double)f_sample)/32768.0);

q0 = g_coef*q1 - q2 + g_sample;
q2 = q1;
q1 = q0;

if(++g_sample_count >= cw_n){
    g_sample_count = 0;
    cw_n = cw_bwa[cw_bwa_index];
    g_current = q1*q1 + q2*q2 -
        q1*q2*g_coef; // ~ energy^2
    q2 = 0;
    q1 = 0;
. . . . .

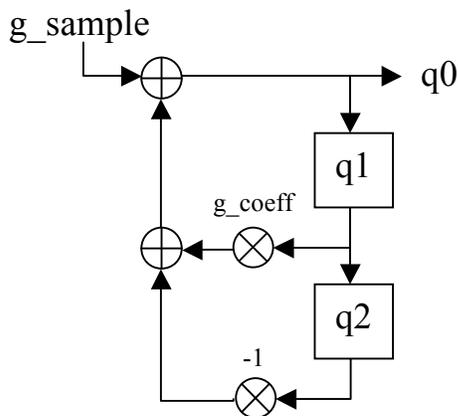
```

The first line simply translates the ADC output from its fractional format (f_sample) to a double-precision floating point format (g_sample). Variables q1 and q2 are initially set to zero.

The Goertzel coefficient (g_coef) determines the center frequency the filter will pass. The modem can be tuned to use center frequencies from 500 to 1,500 Hz. The coefficient is calculated in advance and remains constant as long as the modem is not tuned. When the modem frequency is changed, a new value is calculated and used until the modem is again tuned. The following expression is used to calculate the Goertzel coefficient:

$$g_coef = 2 * \cos(2 * \text{PI} * \text{center_frequency} / 8000)$$

In the code snippet above the `if` statement is executed on each sample in a block of `cw_n` samples. The value of `cw_n` determines the filter's bandwidth. The larger the block is, the narrower the bandwidth will be. The Goertzel filter requires the block size to be an integer factor of the sampling frequency (8,000). The filter bandwidth is equal to the sampling rate (8,000) divided by the block size (`cw_n`). The operator may select from a number of values to vary the filter bandwidth according to operating conditions.



Each block of input samples will generate one filter output value (`g_current`) that is related to the square of the energy within the filter bandwidth centered on the center frequency. A threshold level or “fence” is used to distinguish between MARKs and SPACES in the received signal. Two 1,000-point running averages are maintained for the MARK and SPACE levels and the threshold is set at the average. This allows the average to track and adapt to varying signal strengths. The operator has the option of overriding the fence calculation and to force the threshold up or down.

To filter out some noise, a transition between MARK and SPACE levels will not be registered until it has been steady for 15 sample blocks. Depending on the bandwidth chosen, this adds between one and ten milliseconds of latency to the received signal processing. The filtered filter output is then processed to identify the types of SPACES and MARKs in the input data stream.

SPACE Processing

When CW mode is first entered, it is assumed the receive state is SPACE. The SPACE processor first checks the state of the received signal to determine if it is still SPACE. If it is, it then determines the duration of the current SPACE. This is done by testing the value of a SPACE millisecond

counter. This counter is cleared when the received signal transitions from MARK to SPACE so it will contain the duration of the current SPACE code in milliseconds.

When the received signal changes back to MARK in less than 5 milliseconds, the SPACE is assumed to be the result of noise in the input and is ignored. In this case, the modem goes back to MARK processing as if the transition to SPACE had not occurred.

When the SPACE duration is greater than the `Tcw` count limit for inter-character SPACE, the accumulated MARKs are captured in an eight-element buffer and passed on for translation from Morse code to ASCII.

When the SPACE duration exceeds the `Tcw` count threshold for inter-word SPACE, a space character is displayed on the graphic display. After this, the received signal is only monitored to determine when it transitions from SPACE to MARK. The receive state is then changed to MARK, a MARK millisecond counter is reset to zero, and the next call from the main control loop will be to the MARK processing function.

During SPACE processing, two thresholds or fences are computed and used to detect inter-character and inter-word SPACES in the received signal. The durations of the last ten spaces that are shorter than the current inter-word SPACE are saved and their average computed each time a new SPACE is added to the buffer. The character SPACE fence is set to twice this average. (Remember that intra-character SPACE durations are nominally one `Tcw` and inter-character SPACE durations are nominally three `Tcw`, giving an average of two `Tcw` durations.) The second fence is used to detect inter-word spaces. It is nominally set to five times the average of the last ten intra-character SPACE durations described above. Both of these fences can be adjusted by the operator in increments of tenths of one `Tcw`.

MARK Processing

MARK processing consists of measuring and buffering the duration of each MARK received in the 32-entry MARK buffer and the recalculation of `Tcw` to adapt for changes in received data rates. All MARK processing is done when the function detects the received signal has transitioned from MARK to SPACE. The first check is to make sure the MARK duration is greater than 5 milliseconds. If it is not, it is ignored and the receive state is changed back to SPACE. Note that the SPACE millisecond counter has not been reset and is still measuring the SPACE duration. It is assumed

that any MARK shorter than this is the result of signal interference. (At 99 wpm, the shortest MARK is about 12 milliseconds long.) This action constitutes a noise filter.

After the MARK to SPACE transition has been validated, the MARK duration is noted and compared with a "fence" that is a function of the average of the durations of the last 32 elements buffered. When the duration is greater than the fence, it is recorded as a DAH; otherwise it is recorded as a DIT. After a MARK is classified as a DIT or a DAH, it is buffered in a character buffer. Each entry consists of a code identifying it as a DIT (= 1) or a DAH (= 2). All trailing, unused buffer entries are set to zero. The character buffer can contain up to 8 MARKS. If this limit is exceeded (buffer overflow), the buffer is cleared and a pound sign ('#') is displayed.

The mean deviation of the MARK buffer contents and the average of the last 32 mean deviations are also calculated. The results of these calculations along with the average are used to compute new values for Tcw and the DIT-DAH fence. To avoid skewing these results when long strings of DITs or DAHS are received, all repeated Morse elements after a specific number (the skew limit) are not saved in the MARK buffer and are not included in the calculations. An element is considered to be repeated when its duration is within +/- 12.5% of the duration of the last element in the buffer. An exception is made if four times the absolute value of the mean deviation average is greater than the current MARK average. In this case, the current MARK is accepted regardless of the value of the proceeding MARK duration. This decreases the amount of time required to adapt to rapid changes in received wpm which may occur, for example, when changing frequencies.

The fence used to differentiate between DITs and DAHs is calculated as the average of all the values in the MARK buffer plus the average mean deviation. Ideally, the buffer will contain an equal number of DITs (1 x Tcw) and DAHs (3 x Tcw) and the average mean deviation will be zero resulting in a fence value of 2 Tcw.

Making these calculations at the end of each MARK received enables the receive function to adapt to changes in received wpm. The number of samples used by these calculations determines the speed and accuracy of the results. The lower the number, the faster it will adapt. The higher the number, the more accurate the results will be.

Note that similar analysis is done on SPACE time durations but not as thorough (or as complex) as is done on MARK

time durations. When CW mode is initialized, the MARK buffer and the running averages are preloaded with values representing 20 wpm. The mean deviation buffer is loaded with alternating positive and negative values to give an average mean deviation of zero. This generally reduces the time required for the mode to synchronize with the received CW signal.

Morse Code to ASCII Translation

Each buffered Morse character is translated to ASCII per the table in Appendix A and displayed. As explained above, the MARKS in a received Morse character are buffered as ones for DITs and as twos for DAHs. All unused character buffer locations contain zeroes. Translation is done determining the number of MARKS (non-zero entries) in the buffer and then computing a weighted sum of the buffer contents. The weighted sum is generated by summing the contents of each buffer location multiplied by two raised to the power of the buffer location. The following table shows the multipliers for each buffer position:

Buffer Position:

0x 2 ⁰	= 1
1x 2 ¹	= 2
2x 2 ²	= 4
3x 2 ³	= 8
4x 2 ⁴	= 16
5x 2 ⁵	= 32
6x 2 ⁶	= 64
7x 2 ⁷	= 128

The weighted sum is designed to be unique for any combination of DITs and DAHs in the received Morse character. How the sum is processed depends on how many MARKS are in the character. Note that the Morse code for all letters contains 1, 2, 3, or 4 MARKS. Also, the largest weighted sum for a standard alpha character is 28 ('Q' = --- = 2x8 + 2x4 + 1x2 + 2x1 = 28). This means that for all cases where the sum is less than 29 and there are four or less MARKS in the buffer, the weighted sum may be used to index into the following character string to translate the Morse character to ASCII:

"ETIANMSURWDKGOHVFL*PJBXCYZQ"

The two asterisks represent weighted sums (DIT-DAH combinations) for which there are no corresponding ASCII codes. It is interesting to note that characters were assigned Morse codes based on the frequency of occurrence of the letters in normal text. The characters in the above string,

therefore, occur in order of decreasing occurrence frequency in “normal” text.

Buffers containing five, six, seven, and eight each represent smaller number of characters so they are simply brute-force translated to ASCII. Some of these categories contain prosigns which are translated to two lower-case letters (e.g., Break = -...-.- = “bk”).

CW TRANSMISSION

During transmission, scan codes from a standard PC-style keyboard are translated to ASCII, displayed on the LCD display, and translated to Morse code characters. The modem also provides seven macros that may be inserted in the transmit buffer using the first seven function keys (F1 to F7). Each macro may contain up to 255 characters. The Morse code characters are used to generate base-band audio using a digital to analog converter (MCP4922 DAC) and sent to the transceiver to modulate the RF output.

Key Processing

Serial keyboard data is first processed by a dedicated, 8-bit microprocessor (68HC908QY4) programmed to de-serialize the keyboard data and present it as character data to one of the modem controller’s PIO ports. Each character presented is announced with an interrupt and is pushed into an eight character FIFO buffer. The processing of the keyboard data characters is outside the interrupt routine called from the main control loop whenever the FIFO buffer is not empty.

Keyboard data is referred to as “scan codes” and references keyboard key positions. The meaning of many scan codes is dependent on the states of three special keys (Shift, Ctrl, and Alt) and whether or not the scan code was preceded by a special scan code indicating a two-code sequence. Needless to say, scan code processing is not very straightforward. The modem uses a 132-row by 5-column two-dimensional array of 16-bit integers. Rows are accessed using the scan code; columns are accessed depending on the states of the special keys.

ASCII characters read from the array are pushed into a 512-character FIFO buffer. Some of these characters may not have Morse code equivalents and will be essentially ignored when they are processed for transmission.

Macro Processing

Users can define seven macros of up to 255 characters each that can be inserted in the transmit data queue using the Function keys F1 through F7. Macros are contained in an EEPROM and can be loaded, off line, from the keyboard or through a USB port using a flash drive. Macro character processing is done pretty much as key data processing with one exception. Keyboard data is displayed on the LSD display when the characters are entered into the transmit buffer allowing keying ahead and some editing of the data queued for transmission. Macro data is displayed as it is transmitted and cannot be edited in the transmit buffer.

ASCII to Morse Code Translation

Each ASCII code queued for transmission is translated to a 16-bit descriptor defining the Morse code for that character. Each Morse element is encoded in two-bit fields starting with the most-significant end of the word. The two-bit element codes are interpreted as follows:

00	end of character
01	DOT
10	DASH
11	SPACE of two dot times

Each character is sent separated by a SPACE of one dot time between them so normal character representations end with “11” which generates three Tcw of SPACE between characters. An ASCII space code (0x20) is represented by 1111 1100 0000 0000 which generates seven Tcw of SPACE between words. When the ASCII character is a lower case alpha character it is assumed to be part of a prosign and the ending “11” is omitted and the inter-character SPACE is reduced to a single Tcw.

CW Carrier Generation

CW mode uses the CWID capability built into the original modem and, optionally, used to tag BPSK transmissions. This mode generates the I and Q phase shifts to generate a pure sine wave at the tuned transmit frequency. The encoded Morse code descriptor defines the duration of each MARK in the output. The function is called 8,000 times per second and a millisecond counter is used to time the duration of the MARKs and SPACES. CW transmission speed (wpm) is controlled by the operator.

Abruptly turning a carrier on and off will generate some undesirable side effects like “key” clicks and splatter. To avoid this, the generated sine wave envelope is shaped so

that the carrier gradually turns on and off. This is done by shaping both the start and end of the sine wave for each MARK with a table of constants ranging from 1.0 to 0.0 in 32 steps, generated by a sinc function ($\sin(x)/x$). A single table is used for both ends of each MARK. Leading edge shaping accesses the table in reverse order. Trailing edge shaping accesses the table in the forward direction.

OPERATION

The heartbeat of CW mode is a timer that generates an interrupt every millisecond. The interrupt service routine (ISR) increments the MARK and SPACE counters. The first two are used to time SPACES and MARKS, respectively, during CW receive operations. The general firmware structure is such that the time spent in the ISR is minimized. All the processing functions are called from the main control loop, outside of any ISR. The CW control loop has two function calls. A state variable (`cw_state`) determines which of two CW receive functions is called. One receive function runs during key-up (SPACE) and the other during key-down (MARK).

TUNING

Normally, the transceiver translates received RF to baseband audio referenced to the frequency displayed on the transceiver dial. The modem uses DSP techniques to analyze and demodulate the audio signal and extract the desired information according to the mode the modem is operating in. The modem can process signals in a 2 KHz spectrum of audio frequencies ranging from 500 Hz to 2500 Hz. Tuning is aided with a spectrum display representing the energy detected in each of 128 frequency slices or "buckets". Each bucket represents a 15.625 Hz slice of the spectrum and is displayed as a vertical bar chart or histogram. The height of each of the 128 bars represents the amount of energy detected in the associated bucket. The histogram data is generated by a 512-term, Fast Fourier Transform (FFT) with a sampling rate of 8,000 samples per second.



The spectrum display includes a cursor representing the current frequency of interest. The cursor may be moved in a couple of ways by the operator. During normal operation, the operator sets the transceiver frequency dial to a base frequency and can then tune the modem to cover a range of between 500 Hz and 2500 Hz above the base frequency. The modem also includes an AFC function that will move the cursor to an adjacent bin if that bin has more energy in it than the current one. Tuning can also be done by leaving the modem cursor at a fixed position (e.g., centered at 1500 Hz) and tuning the transceiver to move displayed bars to the cursor.

The normal modem tuning features are available in CW mode; however, the tuning range is limited to a maximum of 1500 Hz and confined to the left half of the display. During operation, the transceiver is tuned to get the desired signal's bucket on the display. This is done by listening to the transceiver audio and watching the modem display as the transceiver is tuned. Fine tuning is then done by moving the cursor to the bucket. The modem tuning controls allow tuning in steps of 10, 5, 2, or 1 bucket using keyboard hot keys. The modem also provides a tuning dial that can be rotated left or right to move the cursor down or up the spectrum display. Pressing the End key on the keyboard will activate the AFC function and the cursor will automatically be moved to the bucket with the most energy within two buckets of the bucket currently pointed to by the cursor.

It will not generally be apparent to the operator but the dial frequencies on the two transceivers will be off by the modem tuning frequency when the remote transceiver is operating in CW mode. For example, if the local transceiver is tuned to 14,060.00 KHz and the modem is tuned to 800

Hz, the remote transceiver will be tuned to 14,060.80 when its tuning indicator indicates “on frequency”.

CW TRANSMISSION

Status Display

In transmit state, the spectrum display is cleared and replaced by the status information displayed on the second and third lines of the LCD display:



Where “WPM” is a two-digit representation of the current wpm (words per minute) setting. The wpm setting may be varied between 5 wpm and 99 wpm by turning the tuning knob when the above status is displayed (modem in the CW transmit state).

Transmit Side Tone

As an added feature, the modem’s beeper may be used to generate a side tone so the operator can listen to the Morse code being transmitted. The transmit side tone generator is toggled on and off by pressing and releasing the ESC key while in CW transmit state. The side tone frequency can be adjusted by holding the CONTROL key down while turning the tuning knob. The frequency can be adjusted to any of nine frequencies between 400 and 2,000 Hz. There is no volume control; however, the frequency response of the “speaker” is such that some frequencies sound much louder than others. Changing the side tone frequency offers some control of the side tone apparent volume. (My preferred side tone frequency is 1333 Hz.)

The three or four digit value on the third line is the current side tone frequency in Hertz. Note that side tone frequency is only displayed in transmit state when the side tone is turned on. The on-off state of the side tone, the side tone

frequency, and the wpm will be saved with a saved configuration and automatically restored each time the modem is powered up.

Keyboard Input

In CW transmit state; the keyboard is forced into Caps Lock mode where all keyed alpha characters will be entered as upper-case letters unless a SHIFT key is pressed. All alpha characters keyed while the SHIFT key is depressed will be entered as lower-case letters. Lower-case letters are transmitted as prosigns by following the last CW element with a single (one Tcw) inter-element SPACE instead of the normal (three Tcw) inter-character SPACE. For example, keying Shift-B, Shift-K, and Space will result in the prosign **bk** being transmitted as “-...-”.

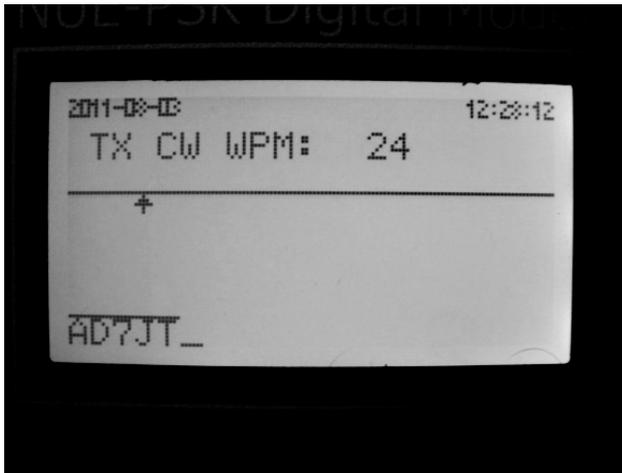
During CW transmission, keystrokes are queued as ASCII characters in a first-in-first-out (FIFO) buffer until a code representing a space code is entered. At this point, characters are read, one at a time, from the FIFO, translated to Morse characters, and transmitted up to and including the space character which is transmitted as an inter-word SPACE. Sending words as a contiguous block of Morse characters avoids pauses in the input keying causing the insertion of unwanted, inter-word SPACES in the transmitted text. The inter-word SPACE duration is only specified as minimum so longer pauses between words generally do not cause a problem (assuming the delay does not exceed the receiving operator’s level of patience). This approach also allows error correction and editing within words. When the operator continues keying during transmission, space characters are counted as they are entered into the FIFO. This count is decremented each time a space character is removed from the FIFO. As long as the space count is greater than zero, transmission will continue. Note that Prosigns follow the same rules as other characters and will not be transmitted until they are followed by a space character in the FIFO.

The transmit FIFO is 256 characters deep and is a circular or “ring” buffer. That means that after the last character position in the buffer is loaded, the next character will be loaded in the first character position of the buffer. There is no overrun check, if buffer input (keying or macro) gets 256 characters ahead of the transmitted data stream, the 256-character contents of the buffer will be lost and will not be transmitted. It is recommended that the operator not have more text queued for transmit than will fit in the transmit text area of the LCD display. This corresponds to about 80-

character type-ahead. (The full 256 character FIFO capacity is required for queuing macros for transmission.)

Transmit Text Display

Keyed CW text characters are displayed on the lower four lines of the graphic LCD display as they are entered even though there will usually be a time delay until they are transmitted. Keyed characters are displayed on the LCD with a continuous over line. For example:



If this is all that is keyed, nothing will be transmitted. The first character is only transmitted after a space code is keyed and entered in the buffer. As each character is transmitted, the over line is removed to indicate to the operator what has been transmitted and what remains queued to be transmitted. The modem maintains a count of the number of space codes in the transmit buffer that have not been transmitted. This count is incremented each time a space code is inserted into the transmit buffer and is decremented each time a space code is read from the buffer. Transmission will continue as long as the space count is greater than zero. When the space count is decremented to zero, transmission will pause until another space code is inserted in the buffer.

Since CW text is essentially transmitted in words, it is not possible to backspace in (or into) a word that has started to be transmitted. In other words, the backspace key will not work when the space count is equal to one. It will work when the space count is equal to zero indicating there is only one word to be transmitted in the buffer and the trailing space has not been entered. It will work when the space count is greater than one indicating there are at least two words in the buffer and the last one has not started to be transmitted. It is not possible to backspace into a word

on the display that has already been transmitted (no over line).

The hot-keys Alt-M, Alt-T, and Alt-S may be used to insert “My Call”, “Their Call” and a serial number (see Section 5.3) in the transmit buffer. In these cases, the characters are entered as if the operator keyed them and they are displayed with the over line. They will not be transmitted until a following space code is keyed. The call signs must be entered (Ctrl+M and Ctrl+T) with upper-case letters to avoid generating unwanted prosigns. Spaces in a call sign are not allowed and may cause erratic behavior.

CW Macros

When macros are played, the characters are displayed as they are transmitted so over lines are not used. They also do not require an ending space code to transmit the last word in the macro. The CW macros are a different set of macros than the ones used by the other, digital modem modes. They are accessed normally (F1 – F7) but only when the modem is in CW mode. The normal Ctrl-key combinations may be used to insert <TXON>, <TXOFF>, <MYCALL>, <THEIRCALL> and <SERIALNO> codes in macros. Note that the data is case sensitive. Normal alpha text must be entered as upper case letters. Lower-case letters will generate prosigns as described above. Each of the seven CW macro buffers can hold up to 255 characters.

CW RECEPTION

As previously described, during CW reception, the transceiver audio is monitored over a specific frequency spectrum (500 to 2500 Hz). A Goertzel filter is used to isolate the MARK-SPACE signals being received at the selected bin of the FFT process. The output of the Goertzel filter is processed by one of two functions. Which is used depends on the last noted state of the received signal. These functions time the duration of the received MARKS and SPACES and determine the type of elements and SPACES received. The MARK types (DIT and DAH) are buffered until an inter-character SPACE is received. The buffered MARK types are interpreted and translated to a Morse character or a prosign and displayed on the lower half of the graphics display as described above.

Receive Text Display

In receive state, received and decoded CW text is displayed on the bottom four lines of the graphic LCD display. Each line displays up to 20 characters. As in other modes, lines

are scrolled up when the end of the bottom (eighth) line is reached. Characters are displayed as they are decoded. Normal text is displayed as upper-case letters; recognized prosigns are displayed as pairs of lower-case letters. Recognized prosigns are listed in Appendix C.

Unrecognized (invalid) Morse characters are displayed as a single asterisk (*). When a character overflows the 8-entry MARK buffer the buffer is cleared and a pound sign (#) is displayed.

After each character is displayed, the current wpm is calculated and displayed at the center of the top line of the graphic LCD display. Note that the wpm will not be displayed until at least one character has been received and Tcw can be calculated.



Figure 5: CW transmit mode. Note the ending prosign KN.

Spectrum Display

The standard FFT display covering the range of 0.5 to 2.5 KHz (above the frequency the receiver is set to) is displayed on the top half of the graphic LCD display along with the optional RTCC date and time information. In receive state, the current, computed receive wpm is displayed in the center of the top line

In receive mode, the right half of the spectrum display can be used to display and/or modify/edit a number of parameters. In all cases, pressing Enter will clear this portion of the display. Switching to transmit mode will also clear this portion of the display but it will be refreshed after switching back to receive mode. The refreshed screen will be displaying the same parameter(s) that was displayed when the modem last switched to transmit mode.

Receive Decode Parameters

There are five parameters affecting the decode operation which may be adjusted within limits by the operator. Two of the parameters define the thresholds for recognizing a SPACE as being inter-character or inter-word. These thresholds are expressed in Tcws. The third parameter specifies the number of (near) equal, consecutive MARK durations that will be accepted and used for timing calculations. The fourth parameter is the Goertzel filter threshold level or fence between MARK and SPACE levels. The fifth parameter is the bandwidth of the Goertzel filter. These parameters and their range limits are as follows:

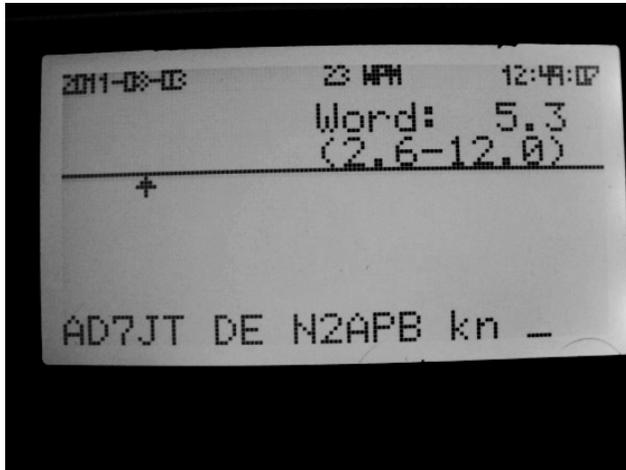
- Inter-character SPACE (Char): 1.6 to 6.0 Tcw with 0.1 Tcw precision
- Inter-word SPACE (Word): (Char + 0.5) to 12.0 Tcw with 0.1 Tcw precision
- Skew Limit (Zkew): 2 to 6 MARKs
- Goertzel threshold (Th): 200 to 64,000 in steps of 200
- Bandwidth (BW): 100 to 1000 Hz, nine selections

These parameters may be displayed in CW receive state by the operator depressing any one of the first letters in the parameter name ('C', 'W', 'Z', 'T', or 'B'). When one of these characters is depressed, the right half of the spectrum display area is used to display the parameter name abbreviation, the allowed range, and the current value. Received data continues to be displayed on the lower four lines of the LCD display.

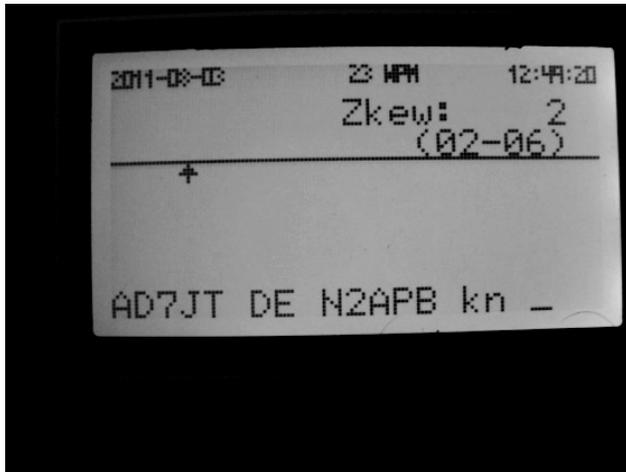
The Char display is as follows:



The Word display is as follows:



The Zkew display is as follows:



The Th display is as follows:



The lower four lines of the display continue to display received CW text and the FFT spectrum display continues to operate.

Changes to the displayed parameter are made with the '+' and '-' keys. The plus will add 0.1 to Char or Word, 1 to Zkew, or 200 to Th. The minus key will subtract 0.1 from Char or Word, 1 from Zkew, and 200 from Th. The range allowed for the parameter is displayed on line three below the parameter name. The minimum for Word is computed to be 0.5 greater than the current value of Char. When the operator tries to change a parameter to a value outside the displayed range, the parameter is restrained to the displayed limit. When Char is changed, the minimum limit for Word will be changed accordingly and if the changed Char value is less than 0.5 below the current Word value, the current Word value (and minimum) will be changed to the Char value plus 0.5.

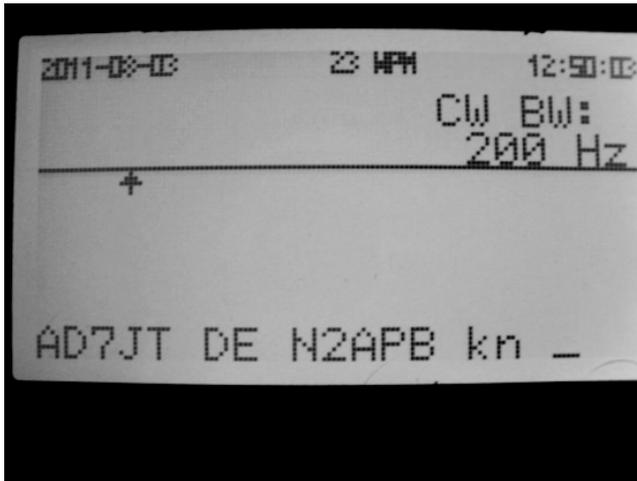
The significance of Char, Word, and Zkew should be obvious from the above discussion. The significance of Th is a little more obscure. The Goertzel filter a fence used to determine whether the received signal is a MARK or a SPACE is computed as previously described. The units used are not standardized so only relative values are used. The energy level computations tend to range between 100 and 60,000 units and are affected by the AGC action of both the transceiver and the modem. Normally, the Goertzel function calculates the fence value based on two running averages, one for high levels (above the current fence) and one for low levels (below the current fence).

When the operator presses a 'T', the current threshold fence value is displayed. Since the display is constantly updated, the operator can watch how the threshold is changing with incoming signal strength. When the operator presses a '+' or '-' key, the fence value calculation is suspended (is locked) and the fence value comes under control of the operator until the Enter key is pressed. When the operator presses the Enter key in receive state, the fence value is unlocked and is again calculated by the Goertzel filter function.

The Char and Word values are also continuously calculated by the modem. When one of these is selected, the displayed value will be updated continuously. When the operator presses a '+' or '-' key, the value is locked and comes under control of the operator until the Enter key is pressed. Note that any or all three (Char, Word, and Th) may be locked at

the same time. Zkew and CW BW are always locked and under control of the operator.

The BW display is as follows:



The CW BW display allows selection of one of nine bandwidths for the Goertzel filter. When the operator presses a 'B', the current bandwidth is displayed. Pressing the '+' or '-' key will move up or down through the available bandwidths. The following options are available:

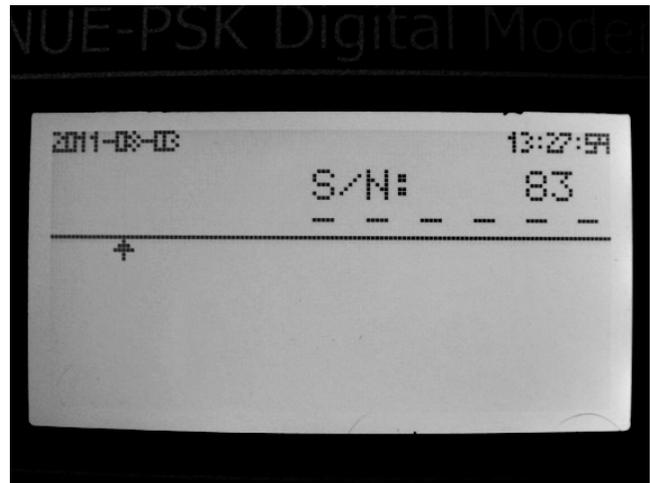
- 100 Hz
- 125 Hz
- 160 Hz
- 200 Hz (Default)
- 250 Hz
- 400 Hz
- 500 Hz
- 800 Hz
- 1000 Hz

Note that the narrower bandwidths may reduce the top keying speed (wpm) the modem can copy. The bandwidth is not saved in the EEPROM nor is it saved or restored with the writing or reading of the CONFIG.TXT file. Each time the modem is turned on, the bandwidth is set to the default value of 200 Hz..

Pressing the Enter key will return the upper portion of the LCD display to the normal FFT spectrum display. At this point, any locked parameters will be unlocked and will again come under control of the modem. Note that none of the keys listed in this section are case (or shift) sensitive and the shift key is ignored. This includes the '+' (or '=') and '-' (or '_') keys.

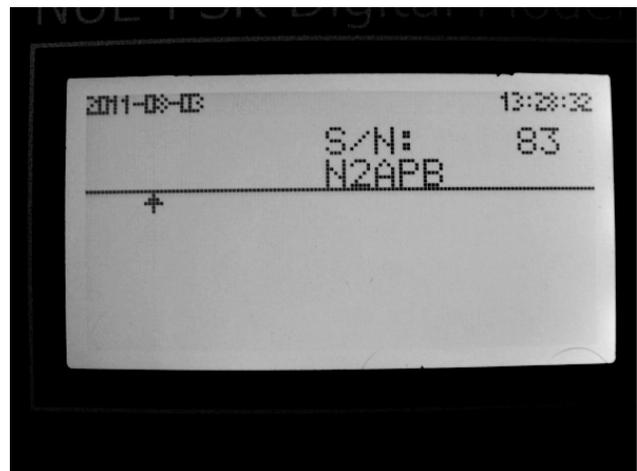
Serial Numbers

Many amateur radio contests require a serial number in the response. Serial number support has been added to the modem. A three-digit serial number is maintained in EEPROM that can be incremented, decremented, and reset to 001 by the operator. There are two displays used to show and control serial numbers. The first display is activated by pressing the 'S' key in receive state and shows just the serial number:



The serial number can be advanced by on pressing the '+' key or decremented by one pressing the '-' key. Pressing the Home key will reset the serial number to 1.

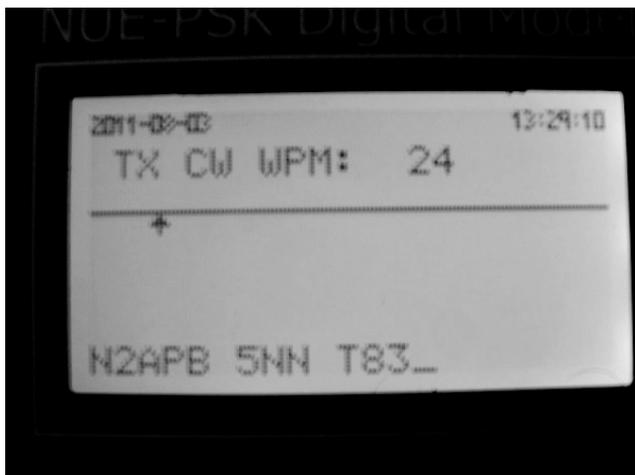
The second display is activated by pressing 'X' in receive state and adds Their Call to the display:



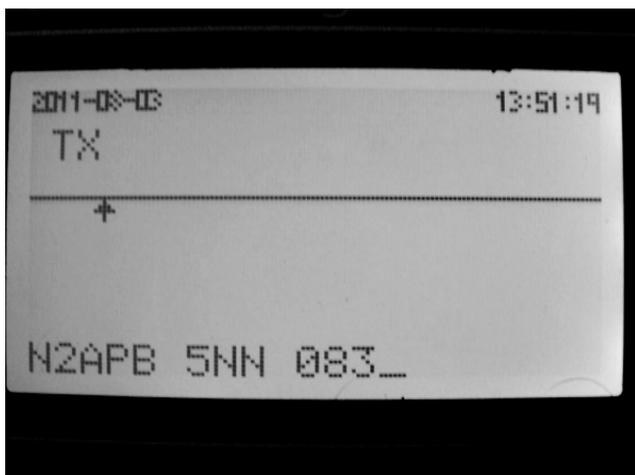
The serial number can be advanced and decremented as with the first display but the Home key will not reset it to one. Pressing any other characters will be entered on the

Their Call line. Pressing the Home key will clear Their call and the backspace key can be used to edit the call.

The current serial number can be entered into the transmit queue by pressing Alt-S or Alt-X, Their Call can be entered into the transmit queue by pressing the Alt-T. Tags can be entered in macros to insert the serial number and Their Call that is current when the macro is played.



Note that, in CW mode, leading zeroes are replaced with 'T' which is a common practice among hard-core contesters. These same features are also available in other modem digital modes (e.g., BPSK). In any mode other than CW, leading zeroes are transmitted as zeroes.



CONFIGURATION

A number of modem configuration parameters are used in support the CW mode of operation. These parameters can be saved in the modem's EEPROM so they will be reloaded each time the modem is powered up.

CONFIG.TXT FILE

NUE-PSK modem equipped with the USB option card uses a special file named "CONFIG.TXT" to allow the user to save/edit/reload various modem control parameters and macro strings for the memory buffers. Please refer to the modem manual for a full description of the file. The format has been extended to include several CW-specific entries, including one that configures the side tone frequency for both the RX and TX side tones. The parameter value for this line may have any integer value between 2 and 10. The side tone frequency is equal to 4000/N Hz, where N is the parameter. This gives the following frequency options:

N	FREQ
10	400 Hz
9	444 Hz
8	500 Hz
7	571 Hz
6	666 Hz
5	800 Hz
4	1000 Hz
3	1333 Hz
2	2000 Hz

KEYER INPUT

A related capability that is useful (but not required) for CW mode operation is that of Keyer Input. A spare I/O line into the modem's dsPIC processor is used as an input to allow an external keyer to be used as input to the CW decode logic. This input is ORed with the Goertzel filter output. To use this feature, one would wire a keyer output to this input. When grounded, this line commands a key down (MARK) condition. This feature was added primarily as a test and debug aid for developing the CW decode logic but it could have some value as a CW training and / or demonstration aid. Future firmware versions may incorporate a full-blown keyer that can be used for transmission too.

IN RETROSPECT ...

The sum total of technical achievements made in the course of doing this project yielded what we believe to be a superior CW reader / keyboard function as implemented in the NUE-PSK modem:

- Goertzel filter to detect the tone
- Morse-to-ASCII conversion
- Automatic speed detection & tracking
- AFC (to simplify tuning and track drifting signals)
- Tuning indicators
- Full user control of parameters affecting CW decoding performance
- Open Source implementation

Even though both of us authors are avid CW operators using the traditional keyer transmit and *between-the-ears* decoding, the modem is quite a pleasure to operate. Keyboard entry for sending Morse characters, and the LCD display of the Morse characters is quite consistent with the operation of other digital modes, and the CW receive speed can be greater, and oftentimes more accurate, than what may be conventionally copied. When coupled with the NUE-PSK's USB logging capabilities, and with decoding speeds up to 50 wpm, the modem really shines during contests.

We also believe this capability will help make operators with limited experience/skills in Morse code more effective and more amenable to using to the "first" mode in ham radio ... CW!

REFERENCES

1. **NUE-PSK Digital Modem**, <http://www.nue-psk.com>
2. **Video of CW reception on NUE-PSK modem** – <http://www.nue-psk/cw/index.html>
3. **CW Robot** -- Rob Frohne, KL7NA, <http://www.tapr.org/pdf/DCC2010-RoboticRadio-KL7NA.pdf>
4. **David Bern, W2LNX** -- "A PS/2 Keyer: Using a Keyer Paddle to Emulate a PS/2 Keyboard and Mouse", QEX May/June 2010
4. **DSP Short Course** -- Rick Meuthing KN6KB, delivered at the 29th Annual ARRL and TAPR Digital Communications Conference. PDFs of the slides are publicly available ...

Session 1: <http://www.tapr.org/pdf/DCC2010-DSPcourse-session1-KN6KB.pdf>,

Session 2: <https://www.tapr.org/pdf/DCC2010-DSPcourse-session2-KN6KB.pdf>,

Session 3: <https://www.tapr.org/pdf/DCC2010-DSPcourse-session3-KN6KB.pdf>,

Session 4: <https://www.tapr.org/pdf/DCC2010-DSPcourse-session4-KN6KB.pdf>.

An inexpensive DVD is available from TAPR containing the video presentation as well ... great course, great deal!

5. Juha Niinikoski, OH2NLT – Discussions concerning Goertzel function coding techniques.

6. Milton Cram, W8NUE – Discussions concerning sample rate processing techniques on the NUE-PSK modem platform.

ABOUT THE AUTHORS

Dave Collins, AD7JT was first licensed as WN0LSH in 1952 at the age of 13 and then as W0LSH in 1953. He graduated from the University of Minnesota with a Bachelor of Physics degree in 1961 and spent over 40 years in the computer industry doing hardware and software development and engineering management. Along the way his license had lapsed but he maintained a strong interest in amateur radio. He retired and moved to Arizona in 2005, passed all four elements of the FCC license exam in one sitting in 2006, and was relicensed as AD7JT. He mainly operates CW on 20M (DXCC in 2008) and has strong interest in SDR and amateur radio projects using microcontrollers and FPGAs. AD7JT may be reached at ad7jt@dnbrealty.com or by mail at 22111 N San Ramon Drive, Sun City West, AZ 85375.

George Heron, N2APB has been a ham for 45 years and is currently a cyber security professional in the Baltimore-DC region. He is a leader in the NJQRP and AmQRP Clubs, and has co-designed various projects for the QRP and homebrewing communities: Micro908 Antenna Analyzer, SDR Cube Transceiver, NUE-PSK Digital Modem, DDS-60 Signal Generator, HC908 VFO, PSK31 Beacon, and many more. N2APB may be reached at n2apb@midnightdesignsolutions.com or by mail at 2419 Feather Mae Court, Forest Hill, MD 21050.

APPENDIX A. CW CODE/CHARACTER CONVERSION TABLE

CHAR	DEC	HEX	MORSE	PROSIGN
	32	0x20	SPACE	
!	33	0x21		
"	34	0x22	..-.-.	
#	35	0x23		
\$	36	0x24		
%	37	0x25		
&	38	0x26		
'	39	0x27		
(40	0x28	-.-.-.	KN
)	41	0x29		
*	42	0x2A	...--	SK
+	43	0x2B	..-.-.	AR
,	44	0x2C	--...--	
-	45	0x2D	-.-.-.-	
.	46	0x2E	..-.-.-	
/	47	0x2F	-.-.-.	
0	48	0x30	-----	
1	49	0x31	.-----	
2	50	0x32	..----	
3	51	0x33--	
4	52	0x34-	
5	53	0x35	
6	54	0x36	-.....	
7	55	0x37	--....	
8	56	0x38	---...-	
9	57	0x39	----.-	
:	58	0x3A	----...	
;	59	0x3B	-.-.-.-	
<	60	0x3C		
=	61	0x3D	-....-	BT
>	62	0x3E		
?	63	0x3F	..-.-.-	

CHAR	DEC	HEX	MORSE
@	64	0x40	
A	65	0x41	.-
B	66	0x42	-...-
C	67	0x43	-.-.-
D	68	0x44	-..-
E	69	0x45	.
F	70	0x46	...-
G	71	0x47	--.-
H	72	0x48
I	73	0x49	..
J	74	0x4A	.----
K	75	0x4B	-.--
L	76	0x4C	..-..
M	77	0x4D	--
N	78	0x4E	-.-
O	79	0x4F	---
P	80	0x50	.-.-.
Q	81	0x51	--.-
R	82	0x52	..-.
S	83	0x53	...
T	84	0x54	-
U	85	0x55	...-
V	86	0x56-
W	87	0x57	.-.-
X	88	0x58	-.-.-
Y	89	0x59	-.-.-
Z	90	0x5A	--..

APPENDIX B. HOT-KEYS FOR CW MODE

- F10:** Toggles between receive (RX) and transmit (TX) modes.
- C or c:** Inter-character SPACE (char): 1.6 to 6.0 Tcw with 0.1 Tcw precision
Determines the threshold between received intra-character SPACES and inter-character SPACES.
- Z or z:** Skew Limit (Zkew):..... 2 to 6 MARKs
Number of MARKs of equal duration that will be used to calculate thresholds. Consecutive MARKs are considered equal when their durations are within +/- 12.5 % of each other.
- W or w:** Inter-word SPACE (word):(char + 0.5) to 12.0 Tcw with 0.1 Tcw precision
Determines the threshold between received inter-character SPACES and inter-word SPACES. Will be automatically incremented when the Inter-char SPACE parameter is incremented to within 0.5 Tcw.
- T or t:** MARK-SPACE threshold..... 64,000 to 200 with 200 unit precision
Displays or sets the threshold/fence used by the Goertzel filter to distinguish between MARK and SPACE levels.
- F or f:** FFT CW Spectrum display
Displays a numerical representation of FFT buckets 18, 19, and 20.
- + or =:** Increment selected parameter +1
- or _:** Decrement selected parameter..... -1
- Enter:** Clears parameter display.
- Esc:** Toggles RX side tone on and off.

APPENDIX C. RECOGNIZED PROSIGNS

MORSE	PROSIGN	MEANING	DESCRIPTION
.-.-..	al		???
.-.-.	ar	All Right	end of message
.-...	as	wait A Sec	stand by
-...-.-	bk	BacK-to-you/let me in	BreaK
-...-	bt		Separator
-.-...-	cl	CLear	CLosing down
-.-...-	cq		Calling
-.-.-	ct	Commence Transmission	start of message
.....	hh	error	HuH?
..-.-	iq		???
-.-.-	kn	oK, Named-station	Kalled station only
...-.-	sk	Silent Key	end of contact
...-.	sn	Sho' 'Nuff	understood