HamHUD II

User's Manual

Firmware Version: 2.7.xxxx for PIC18F252 (20MHz and 40MHz)

Original document by: Steve Bragg KA9MVA Owasso, Oklahoma
Last Updated: May 15, 2006 by Jason Rausch KE4NYV Roanoke, Virginia

Forward

- The HamHUD is a APRS community project that has spanned many years and has been updated several times to grow with the needs of its users. This document is a full operating manual for the use and operation of the HamHUD. This manual will also document the changes and upgrades the project has gone through.

Introduction

- The current HamHUD II hardware is based around the Microchip PIC18F252 running at 40MHz. First and second generation HamHUD II's are capable of this firmware with a replacement of the original PIC16F876 and 20MHz crystal with the PIC18F252 and a 10MHz crystal (clock speed is internally multiplied by 4). Both items are drop-in replacements and do not require any modifications to the PCB itself. The PIC18F252 does require that it be initially burned with a programmer such as the older Warp13 or newer Olimex MPLAB compatible programmer.

Features

General

- **Connect up** to your Kantronics KPC-3x TNC or any TNC2 compatible clone and GPS
- **Automatic GPS detection**, and on-screen warnings when GPS is not present or out-of-lock Position data can come from a GPS ($GPRMC sentence), or be entered manually.
- **4-line x 20-char display**, ordered as "My Data" (line 1), "Auxillary Display" (line2), and "Others Data" (lines 3 & 4)
- "**Whiz Wheel**" Single knob with button.   Intuitive control of: settings, functions, and data entry
- "**BEEPER**" to indicate incoming messages and mode changes
- **Tricolor LED** indicating XMIT, DIGIPEAT, and RX; Green LED for TNC's "CD" indicator
- **Displays most APRS® data**, including posits, messages, weather, status, GPGGA, GPRMC, TH-D7A / TMD700 / / PIC-E / MIC-E, APRS objects, etc;
- **Handles ANY packet**; prints raw data of any packet it does not understand (no "??KA9MVA" like the Kenwood radios!)
- **Reload firmware and customize settings** using your PC!

APRS-Network-Related

- SmartBeaconing (tm) controls beaconing according to speed, cornering, and network activity.
- **TOTALLY TUNABLE** parameters - **works for anything from automobiles to zeppelins**!
- **'='-style posits** for beacons, or use $GPRMC strings, your choice
- **DigiMeter**^{tm} tells how well you're getting in, and through which digipeaters
- **Status**: select one of 6 editable status strings to transmit at a selectable interval
- **APRS queries** (stores 6, up to 20 characters each). Find other stations, weather stations, IGATES, etc.!
- **Responds to queries**: ?APRS? and ?HUDS?
- **UNPROTO**: Three, 20-character, editable paths. Destination address (e.g., CQ, APHH2, SPCL, etc.) is editable. MYCALL is editable. All UNPROTO changes take effect immediately.
- **Control your TNC!** Send arbitrary commands (stores 6, up to 20 characters each) to adjust other TNC parameters, even on the go!

User Features

- **APRS® messaging**, with 12 outgoing and 12 incoming messages stored. Flashes the LCD backlight and beeps the BEEPER when you have an incoming message. InBox indicator ensures you don't miss your messages! Scrolls messages wider than display.
- **Station list.** 48 entries. All heard stations, and their posits are stored here.. Weather stations with posits in the station list show distance/direction. Message "To" callsign can be selected from station list. Rendezvous with stations in the station list.
- **Stay on top of Weather!** Display temperature and wind data from APRS weather stations. Shows distance and direction to weather stations with posits in the station list. Scrolls weather warnings (if they are sent in your area).
- **Hook Mode** makes station rendezvous a snap!
- **Waypoints**: eight, 9-character names. Just click Mark WP, then choose a waypoint name, and you've marked your position. Rendezvous with your waypoints using Hook Mode.
- **Posits**: displays bearing as textual direction with equally-spaced sectors. (e.g., NNE). Efficient distance calculation: VERY accurate at short distances, less accurate farther out
- **Customize the Display!** "Auxillary Display" (Line2) can show : Hooked station or waypoint, Packet Path, GPS Details, SmartBeacon Details. LCD brightness and contrast can be adjusted.
- **Display units**: Metric or U.S. , your choice (SmrtBcon menu).
- **Icons** for '='-style posits is selectable from a list of named icons. Most common mobile icons are included. (Graphical icons are not displayed, however)

Getting Started

TNC Settings

To operate the HamHUD, the TNC must be set up correctly to communicate with the HamHUD. The HamHUD is compatible with just about every TNC2 clone type TNC on the market, but there are some differences in commands from manufacturer to manufacturer.

Kantronics TNC:

```
ECHO OFF
ABAUD 4800
MONITOR ON
CRSUP OFF
LFSUP OFF
HEADERLN OFF
MCOM OFF
MCON OFF
MHEADER ON
FILTER OFF
MXMIT OFF
PACLEN 0
SCREENLN 0
CD SOFT
UNPROTO APHH2 VIA
WIDE1-1,WIDE3-3
HID OFF
```

AEA/Timewave TNC:

```
ECHO OFF
TBAUD 4800
AUTOBAUD OFF
AFILTER OFF
BBS OFF
BBSMSGS ON
BEACON E 0
CONMODE CONVERSE
HEADERLN OFF
HID OFF
MCON 0
PACLEN 0
UNPROTO APRHH2 V
WIDE1-1,WIDE3-3
XFLOW OFF
XMITOK ON
XFLOW OFF
```

MFJ TNC:

```
ECHO OFF
MONITOR ON
HEADERLN OFF
MCOM OFF
MCON OFF
PACLEN 0
SCREENLN 0
HID OFF
UNPROTO APRHH2 V
WIDE1-1,WIDE3-3
XFLOW OFF
XMITOK ON
```

The UNPROTO path should always be set to what is required by your area's particular network. The standard mobile path is typical WIDE1-1,WIDE2-2 or similar.

KPC-3+ DCD Setup

In order to use the CD light (KPC3+ ONLY), the following must be set:

```
CTRL A MUTE 2
```

Please also see **Appendix A** regarding the proper jumpers also required to use the DCD function.

Using a GPS with HamHUD II

The HamHUD II was designed to accept NMEA data from any serial GPS. NMEA V1.5 and later is accepted. Some changes were made to handle NMEA strings that contain more decimal places than others. The current version also allows for the choice of using either the $GPRMC or $GPGGA strings. The recommended string to use is the $GPRMC.

The HamHUD II uses the display to always indicate the status of the GPS. The status is represented in three different states:

- **NO GPS -** This indicates that either a GPS is not connected to the unit and/or the GPS is not supplying the data required by the HamHUD II..
- **NO FIX** - This indicates the a GPS is connected and the HamHUD II is successfully reading the NMEA data from it, but the GPS has not gained a fix on it's location yet. Some GPS strings are structured slightly different that does not allow the HamHUD II to sense a NO FIX state. This is usually causes by GPS receivers that only send something like ($GPRMC,,,,,,,,,,,,) when it does not have a fix.
- **0@0 -** This indicates that the GPS has gained a fix and the HamHUD II has successfully read the position data to calculate positions, speed and heading with. When moving, the 0@0 will changed to something like 27@195, indicating you are moving at 27 MPH on a heading of 195 degrees.

Why is there a "?" next to displayed distances?

- The HamHUD II always make an attempt to do what it can, with what it has. Anytime the HamHUD II makes a distance calculation, it uses the current Lat/Lon position to determine that distance. If at anytime the HamHUD II makes a distance calculation without a GPS connected or with a GPS connected and NO VALID POSITION LOCK, the HamHUD II will assume that the calculation is an estimate. Since it does not have a validated position, it can only calculate the position from the last known good position. Upon receiving current data, the HamHUD II will remove the "?" from the distance displayed on the display.

RevE Kit Additions:

- A TTL header was added to the latest version to give a direct TTL serial link to the PIC for those who would like to use a TTL only OEM GPS board, such as the Motorola Oncore units.
- Separate +12VDC and +5VDC outputs were included in the interface connector to supply GPS units with either required voltage. The +5VDC power does an excellent job at powering small, non-display GPS receivers such as the Deluo and Garmin OEM "puck" units. The +5VDC is supplied from the main regulator, so by turning the HamHUD II off, you can kill the GPS power, thus allowing you to control the GPS power from the HamHUD II.

User Interface

The Whiz Wheel

The Whiz Wheel is a rotary encoder that is used to change settings and enter data. The encoder has a pushbutton switch in it, which is used as an "OK" or "Enter" key would be on a personal computer.

There are two different actions for the Whiz Wheel button: Single Click and a Double Click. A Single Click is one short depress and release, while a Double Click is two quick presses and releases within about 300 milliseconds of each other.

- **Single Clicking -** Used to select items from lists and confirm settings.
- **Double Clicking -** Used to confirm and enter strings. The cursor MUST be on the last character of a string when the button is "double-clicked".

The Beacon Now Button

The Beacon Now button serves several functions:

- **Beaconing -** At any given time, this button can be pushed to transmit a beacon at the current position. There is no limit to the use of this function.
- **Menu Escape -** When in the menu mode, this Beacon Now button can be used as a quick escape from menu mode.
- **TNC Bypass -** If at anytime you wish to operate the HamHUD without a TNC, the Beacon Now button can be used to put the HamHUD into “TNC Bypass Mode” at the time of power. To do this, make sure the HamHUD is turned OFF. Hold down the Beacon Now button and turn the HamHUD on. You will hear the internal buzzer indicating the button is pressed. Continue the hold the Beacon Now button until the unit’s callsign appears on Line 2.

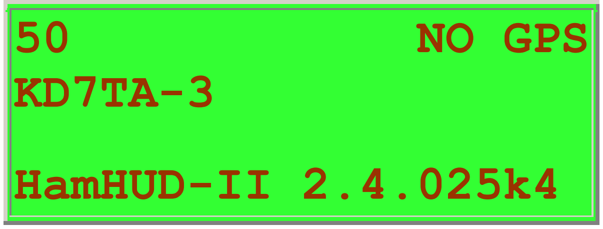
Packet Indicator LED

To indicate what the HamHUD II is doing when transmitting, receiving, ect., a bi-color LED is used. The LED is capable of three color states:

- **Green** - Indicates when you are receiving a packet from another station.
- **Red** - Indicates when you are transmitting a packet.
- **Orange** - Indicates when you are receiving a digipeat of your own packet.

Power Up

When the HamHUD powers up, it waits a few seconds, then tries to talk to the TNC and get its callsign. If it is successful, you will see the display below (your callsign will be shown):



```
50            NO GPS
KD7TA-3

HamHUD-II 2.4.025k4
```

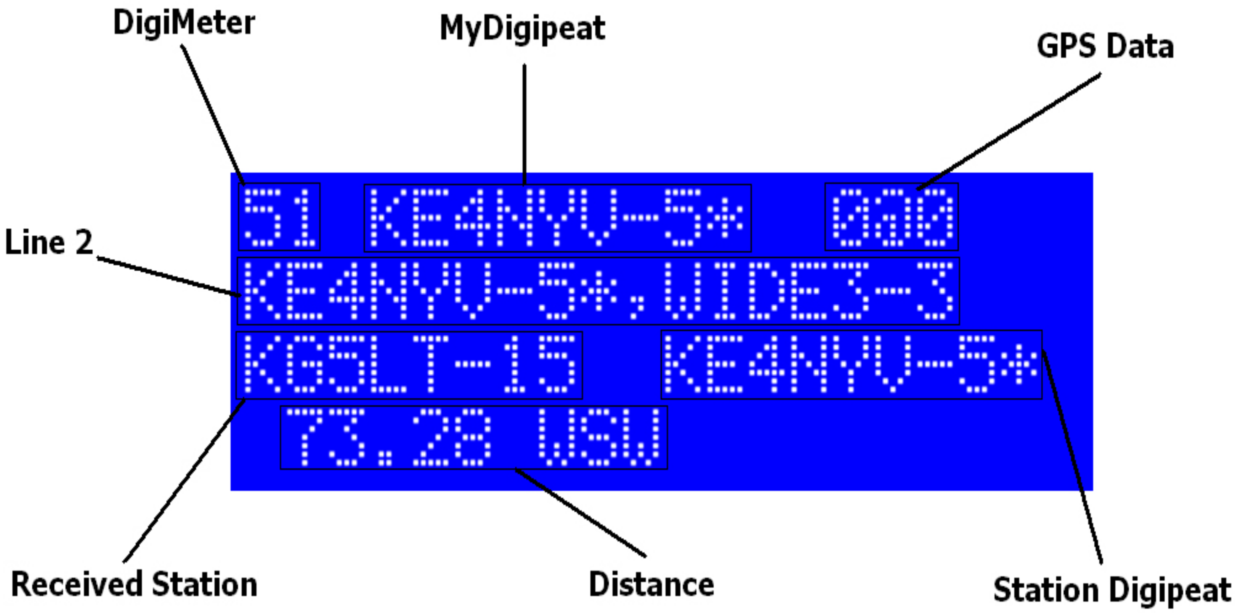
If you are unsuccessful, check that the TNC settings and serial connections are correct.

As soon as the callsign is displayed, the HamHUD is ready to receive packets and activate its menu.

TNC Bypass

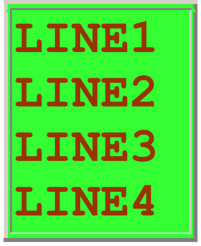
If at anytime you wish to operate the HamHUD without a TNC, the Beacon Now button can be used to put the HamHUD into “TNC Bypass Mode” at the time of power up. To do this, make sure the HamHUD is turned OFF. Hold down the Beacon Now button and turn the HamHUD on. You will hear the internal buzzer indicating the button is pressed. Continue the hold the Beacon Now button until the unit’s callsign appears on Line 2.

The HamHUD Display



Use of the Display

The HamHUD display is a 20-character by 4-line liquid-crystal display (LCD). It is separated into three distinct display areas**:**


LINE1
LINE2
LINE3
LINE4

- My Data (line 1): displays my speed, heading, and packet digipeating data.
- Auxiliary Display (line2): displays selectable auxiliary data: path, SmartBeacon data, GPS data, etc. Also may show "my data" if it exceeds what can be shown on line 1 (example: my transmitted messages, my transmitted status).
- Others' Data (lines 3 & 4): displays interpreted packets from other stations.

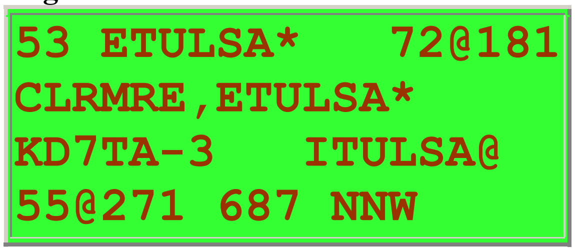
Lines 1 and 2 are also used by the Menu system.

Measurement Units

The HamHUD can display English (Imperial) units, or Metric units. For distances units, statute miles (English) or kilometers (Metric) are used. For speed units, statute miles per hour (English) or kilometers per hour (Metric) are used. For temperature, degrees Fahrenheit (English) or Celsius (Metric) are used. For a description on how to change the display units, see SmartBeacon Parameters.

General and Positions

Figure 1.



```
53 ETULSA*   72@181
CLRMRE,ETULSA*
KD7TA-3   ITULSA@
55@271 687 NNW
```

The DigiMeter

The "53" in the upper left is the "DigiMeter" (tm). The left digit indicates the long-term average conditions of the network you're in, from 0 to 9. The unit start with this digit set to "5", so it goes up if you are into the network. The 2nd digit is the number of digipeated packets from your last transmission.

The DigiMeter is constantly displayed on Line 1. It is updated when a position is transmitted by the HamHUD, or when a digipeated position (originating from your callsign) is received. When the position is sent out, the DigiMeter's right digit will reset to "0". With each digipeat of the last transmitted position, the right digit will increment.

My Digipeater / Destination Call / Inbox

The middle field on Line 1 shows a variety of information. When a position is transmitted, the word "POSIT" shows in this field, followed by the callsign selected as your UNPROTO destination callsign (see the Unproto menu). If any transmission from the HamHUD is digipeated, the callsign of the digipeater (followed by the * digipeat indication) will be shown in this field.

If there are have been received messages since the last time you checked the Inbox (see the Inbox menu), this field will show a message like:

Inbox:3

Indicating the number of messages that are new since the last time you checked the inbox. This reminder will repeat every 10 seconds until you check the Inbox.

My Speed and Heading

The rightmost field on Line 1 is "my" speed and direction. This may come either from my own beacons, or the GPS data coming into the HamHUD. In the latter case, the display is updated as often as receiving other data allows.

This field is continually displayed and updated, as long as incoming GPS data is available. If valid GPS data has not been available for 60 seconds, HamHUD goes back to displaying "NO GPS" in this field until valid GPS data is again available.

Position Data Display

The 3rd and 4th lines show data from "other" stations. In this case, it is showing a packet that was "digipeated" through an IGATE (indicated by "@" instead of "*") Tony's car is doing 55 MPH going west, and he's 687 miles NW of me. Since bearing is always shown as letters, and heading always shown numerically, it's easy to sort them out. All posits are displayed in this format, regardless of protocol. The speed@bearing field is skipped if the posit protocol doesn't include it.

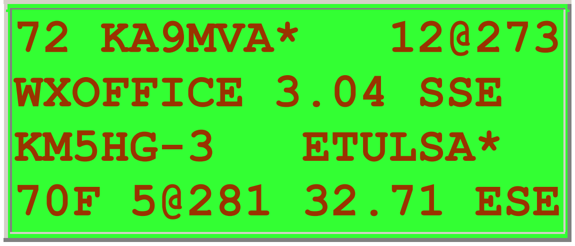
Almost all APRS position formats are supported. Here are the few exceptions:

- Named Items (OBJECTs **are** displayed)
- Grid Square Format
- Compressed Position Format
- NMEA sentences besides $GPRMC and $GPGGA.

Weather Data

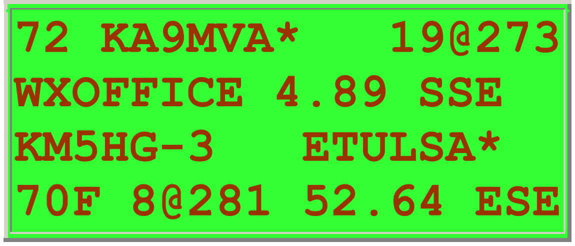
The HamHUD is able to decode most popular weather data formats used with APRS®, including native, Peet Brothers Ultimeter II and 2000, and Davis. These formats include the !! and $ULTW "raw" hexadecimal formats.

Figure 2.
(U.S. Units)



```
72 KA9MVA*   12@273
WXOFFICE 3.04 SSE
KM5HG-3   ETULSA*
70F 5@281 32.71 ESE
```

(Metric Units)



```
72 KA9MVA*   19@273
WXOFFICE 4.89 SSE
KM5HG-3   ETULSA*
70F 8@281 52.64 ESE
```

In this example, Ultimeter-2000 weather is being displayed in line 4. All weather stations, regardless of protocol, are shown this way. When it can, the HamHUD firmware coordinates weather data with position beacons in its station memory. We can see that it's a balmy 70F (for February) in Broken Arrow, Oklahoma (32.71 miles east-southeast of the HamHUD's current position), with light winds (5 knots) out of the west-northwest (281 degrees).

The weather station's distance and bearing will only be shown if the weather station's position is in the Station List (see Station List). Usually, weather stations will transmit their position and weather data close together in time, or in the same packet. HamHUD can handle combined weather / position reports.

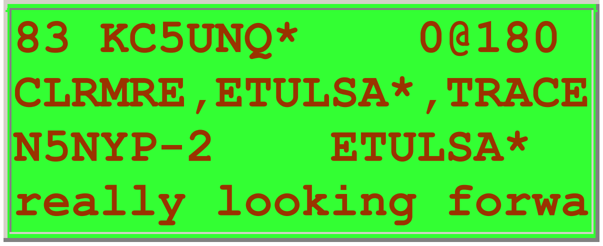
In Figure 2, Line2 is showing the distance and direction to a marked waypoint (see Waypoints), which in this case is the NOAA weather office in Tulsa. While this particular function has nothing to do with the display of weather packets, displaying waypoints is a useful function for severe weather spotters.

APRS® Messages

The HamHUD is able to display messages from APRS® stations, and other HamHUDs. Inbound messages (FROM other stations) are shown in line 4, and outbound (TO other stations) messages are shown in line 2.

Inbound messages cause the HamHUD's beeper to sound for two seconds. (Inbound messages, if they are unique, are also stored in the Inbox.) The beeper sounds after the callsign is displayed, but before the message is displayed, to the give the user time to notice that a message is incoming.

Figure 3.



```
83 KC5UNQ*    0@180
CLRMRE,ETULSA*,TRACE
N5NYP-2    ETULSA*
really looking forwa
```

In the above display, N5NYP has sent a message, which is being scrolled, bank-sign-style, in the 4th line.   His path is currently visible in the auxiliary display (line 2).

Acknowledge Messages (ACKs)

The HamHUD will ACK inbound messages. As soon as the message stops scrolling, HamHUD sends an ACK back to the sending station.

ACKs from other stations are displayed on line 4, as ACKs are considered to be inbound messages. ACKs do not affect the transmission of messages.

Using the HamHUD Menus

Whiz Wheel Basics

The Whiz Wheel is used to adjust settings, and to perform various actions on the HamHUD. The menus are accessed using the Whiz Wheel. The wheel has two actions: *rotation* and *clicking*.

The Whiz Wheel rotates both directions. In general, rotating the knob clockwise moves the cursor to the right, or increases a setting. Rotating counterclockwise moves the cursor to the right, and decreases a setting. The Whiz Wheel has rotation speed filter, to prevent spurious rotation detection. Rotating the Whiz Wheel quickly has little effect, but rotating it decisively at a moderate speed works well.

The Whiz Wheel has a pushbutton switch built into it, and pressing the knob activates the switch. This action is called *clicking*. Clicking the knob twice in a short instant of time (< 250 milliseconds) is called *double-clicking*.

Cursors

There are two cursors that the HamHUD uses, the Under bar and Block cursors.

The *Under bar Cursor* (looks like an underscore _), indicates that the character above it can be edited (changed) with the Whiz Wheel. The cursor will stay in place until the menu button is clicked. This indicates to the HamHUD that the user is finished editing that character, and is ready to go on to the next character.

The *Block Cursor* (looks like a filled-in black square □) indicates that character editing is NOT in process, and that the cursor (or setting) will be moved with the Whiz Wheel.

Editing Strings

In many places, the Whiz Wheel is used to edit a string of arbitrary characters, such as a text message, a callsign, or an UNPROTO path. Everywhere a string of arbitrary characters is edited, the HamHUD acts the same way.

Usually, the string to be edited will already have some default value. The HamHUD will display the current value and move the Block Cursor to the end of the string. Turning the Whiz Wheel will move the cursor. To edit a character, the user must move the Block Cursor over it, and click the Menu Knob. The cursor will change to the Under bar, and changing the Whiz Wheel will change the character above the cursor.

A character being edited changes in a specific way using the Whiz Wheel. Turning the Whiz Wheel to the left IMMEDIATELY after the cursor changes to Under bar will cause the character to change to a lowercase 'a'. This allows alphabetic messages to be entered quickly, without going through all the special characters. Turning the Whiz Wheel to the right will cause the character to change in the ASCII sequence, from the space character through the grave accent (`). If the Whiz Wheel continues turning after the grave appears, the character will recycle back to the space character.

Editing with the Whiz Wheel is fairly fast, because the cursor stays in "character edit" mode until a "click" occurs without rotation. The cursor advances automatically to the next location, and the underbars "character edit" cursor appears.   This means an entire sentence can be entered without returning to "move" cursor mode.

As a memory aid, when the cursor advances to the next location, HamHUD starts the character sequence with the last character you used in the previous position. This allows you to quickly "spell out" a string with the Whiz Wheel.

Strings can only be edited as wide as the display,. The display is 20 characters wide.

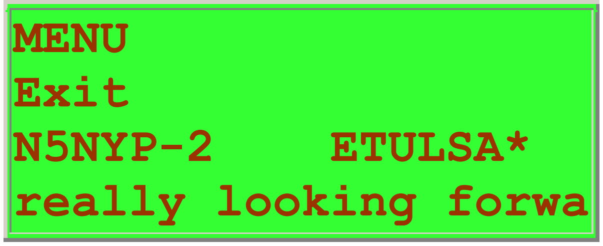
To accept a string, the Block Cursor must be placed ON THE LAST CHARACTER of the string that is to be accepted. Then, a double-click of the Whiz Wheel accepts the string.

NOTE: **ALL callsigns** (including UNPROTO, MYCALL, message to-calls, etc.) **MUST be entered in UPPERCASE**. This is not HamHUD's fault; the AX.25 protocol specifies that callsigns are UPPERCASE only.

Activating the Menu

When the HamHUD's Whiz Wheel is clicked, the following appears on line 1:

Figure 4.



```
MENU
Exit
N5NYP-2    ETULSA*
really looking forwa
```

Having **Exit** as the default menu item allows the user to dismiss the menu with another click of the Whiz Wheel. The Beacon Now/Abort button may also be used to exit the Menu at any time.

Selecting a Top Level Menu Item

Turning the Whiz Wheel to the right causes the next menu item to appear. The menu items appear in this order:

Exit, Message, Inbox, Mark WP, Query, Posit, MyIcon, Unproto, Line 2, SmrtBc, Brite, TNC

Each of these menu items (except **Exit**) has its own parameters or submenus, which will be described in following paragraphs.

Beacon Now / Abort Button

Sometimes a menu item or a submenu will be mistakenly activated. To abort any menu item in progress, press the red button (same as Beacon Now button). The menu will exit, and the usual Line 1 and Line 2 displays will return.

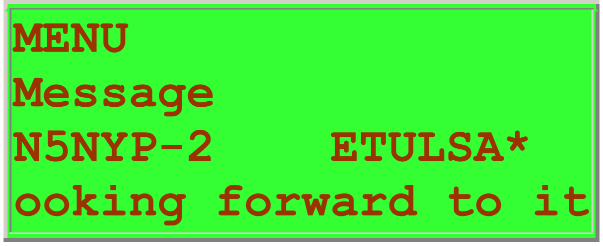
When the menus are not active, this same button will cause a posit

to be sent immediately.

Message: APRS$^{\rightarrow}$ Messaging

Activating the SndMsg top-level menu item initiates an APRS$^{\rightarrow}$ message. This menu item will be illustrated through sending an example message to N5NYP-2. In the example, N5NYP-2 has already sent a message to the HamHUD, and now we will respond to it.

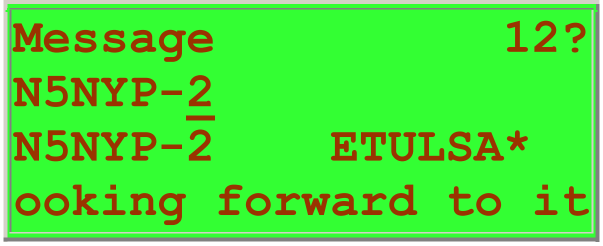
Figure 5.



```
MENU
Message
N5NYP-2     ETULSA*
ooking forward to it
```

Clicking brings up the following display

Figure 6.



```
Message          12?
N5NYP-2
N5NYP-2     ETULSA*
ooking forward to it
```

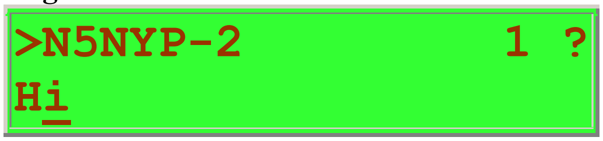
The HamHUD is now asking for the destination callsign. N5NYP-2 was the last callsign received, so his callsign is placed in the destination callsign automatically. If I want his callsign, all I have to do is click the Whiz Wheel. Or, I can turn the Whiz Wheel and select another callsign from the Station List.

NOTE: **ALL callsigns** (including UNPROTO, MYCALL, message to-calls, etc.) **MUST be entered in UPPERCASE**. This is not HamHUD's fault; the AX.25 protocol specifies that callsigns are UPPERCASE only.

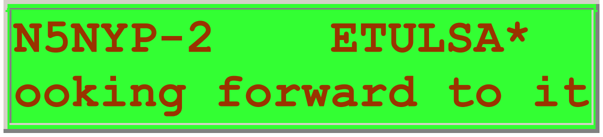
After clicking, the HamHUD's cursor changes from the blinking block to the under bar. Also, the question mark (?) on Line 1 changes to an "E", indicating that HamHUD is inviting me to edit the callsign. It is possible, then, to send a message to a station that is not in the Station List.

When the editing is finished, a double-click brings up the prompt to choose a "canned" message to start with.

Figure 7.



```
>N5NYP-2         1 ?
Hi
```

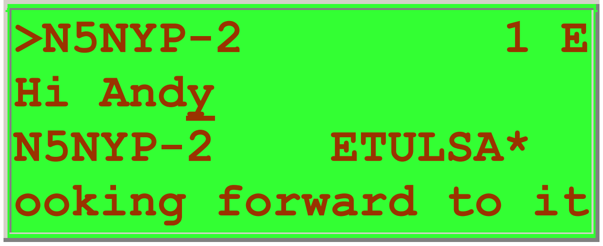


```
N5NYP-2    ETULSA*
ooking forward to it
```

There are 12 canned messages to choose from, each of which can be up to 20 characters. (The messages can be customized from the PC; see Customization.)

In this example, the "Hi " canned message is chosen by turning the Whiz Wheel until it appears, then clicking.

Figure 8.



```
>N5NYP-2         1 E
Hi Andy
N5NYP-2    ETULSA*
ooking forward to it
```

For this example, "Andy" was entered using the Whiz Wheel. After placing the cursor on the last character of the message, double-clicking sends the message.

Resending Messages

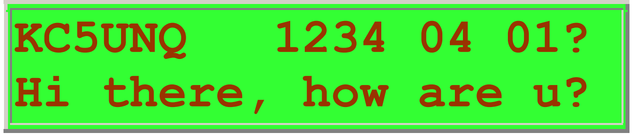
Messages are sent by the HamHUD only once. To send a message again, select it from the "canned" message list , and send it. The resent message will have a different message number than its original, which is slightly different behavior than most APRS→ programs. It is, however, useful for the other station to see which message actually made it through.

InBox: Received Message List

The Inbox stores the 1st 20 characters of your recently received messages. Up to 12 messages are stored.

The Inbox stores the message text, the sending station's callsign, the UTC time, and the message's "Ack Tag". The Ack Tag and message callsign are used to ensure that only unique messages are stored in the Inbox.

A typical Inbox display would be:



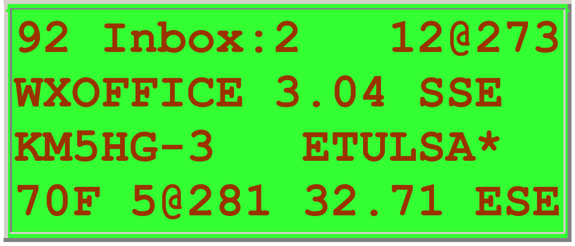
```
KC5UNQ   1234 04 01?
Hi there, how are u?
```

This indicates that the first message (#1) came from KC5UNQ, at 12:34 UTC, at the Ack Tag was 04. The message text is on line 2.

To look at other messages in the inbox, scroll down with the Whiz Wheel. To exit the Inbox without replying to a message, press the Red Button.

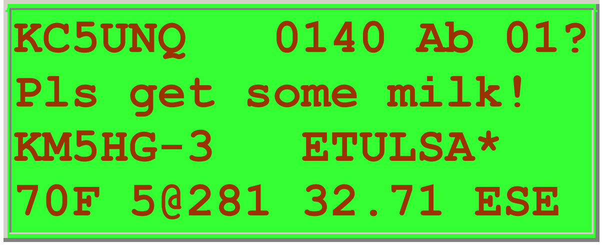
Replying to a Message

This is where the Inbox is very useful. Suppose you are on a shopping trip. You leave the car to shop, and when you return, line 1 is blinking every 10 seconds, in the digipeater field:



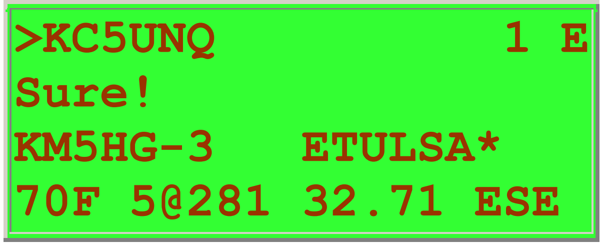
```
92 Inbox:2   12@273
WXOFFICE 3.04 SSE
KM5HG-3   ETULSA*
70F 5@281 32.71 ESE
```

This means you have missed two messages since you last entered the Inbox. You select the Inbox menu item, and are greeted with:



```
KC5UNQ   0140 Ab 01?
Pls get some milk!
KM5HG-3   ETULSA*
70F 5@281 32.71 ESE
```

To reply to the message, simply click the Whiz Wheel button. You might reply:



```
>KC5UNQ          1 E
Sure!
KM5HG-3   ETULSA*
70F 5@281 32.71 ESE
```

The callsign is automatically filled in for you, from the Inbox callsign. You simply select and/or edit a message, and double-click on the last character to send it.

Mark WP: Mark Current Position as a Waypoint

This feature allows navigation back to a particular position. When the Mark WP menu item is selected, HamHUD immediately stores the current GPS position. Then, it asks the user to select and/or edit the waypoint name. The position is stored under the waypoint name. The waypoint can be "hooked" using the Line2->Hook menu item.

Query: Send an APRS→ Query

Initiating Queries

This allows the HamHUD to see which stations are within range, and where they are.

Initiating this menu item brings up the default "?APRS? query, and allows the user to edit it. There are 6 "canned" queries (customizable from the PC), which are also editable in the HamHUD.

Other queries (or even arbitrary data strings, regardless of the APRS protocol) may be entered. Double-clicking with the cursor on the last character t of the query causes the HamHUD to transmit it.

Responding to Queries from Other Stations

The HamHUD will automatically respond to ?APRS? and ?HUDS? queries by transmitting a posit as soon as possible.

Posit: Edit Stationary Position Report

In the absence of GPS input, it is still possible for HamHUD to send position to other stations. Editing the APRS posit string in the Posit menu item allows operation from a hotel, a fixed digipeater site, etc.

The Posit is **EXACTLY** 20 characters long, and looks like this:

```
|-------------------------- This character should be '=' for a station with
messaging, or '!' for an unattended digipeater.
|         |---------|------ These are the icon table select and symbol
characters.
v         v         v       They will be automatically changed by the Icon
menu.
=3605.06N/09550.38W>
^^^^^ ^^^ ^^^^^ ^^^
| | |  | | |  |  ||_______ Western or Eastern Hemisphere designator.
| | |  | | |  |  |________ Hundredths of minutes of Longitude
| | |  | | |  |___________ Minutes of Longitude
| | |  | | |______________ Degrees portion of the Longitude
| | |  | |________________ Northern or Southern Hemisphere designator
| | |  |__________________ Hundredths of minutes of Latitude
| | |_____________________ Minutes of Latitude
| |_______________________ Degrees portion of the Latitude
|__________________________  = indicates an APRS position.  THIS IS REQUIRED.
```

NOTE: THE POSIT **MUST** BEGIN WITH AN EQUALS OR EXCLAMATION CHARACTER, FOR OTHER APRS STATIONS TO CORRECTLY INTERPRET IT.

The Latitude consists of six parts:

1. 2 digits for the number of Degrees Latitude. Range: 00-90 degrees.
2. 2 digits for the number of Minutes Latitude. Range: 00-59 minutes.
3. Decimal point. Must be there in that position.
4. 2 digits for the fractional part of minutes of Latitude. Range: 00-99
5. 1 character representing the hemisphere N or S
6. An icon table select character (usually "/", but see the APRS Specification for details). Required. **THIS CHARACTER IS AUTOMATICALLY CHANGED BY THE ICON MENU**.

The Longitude consists of six parts also

1. 3 digits for the number of Degrees Longitude. Range: 000-180 degrees.
2. 2 digits for the number of Minutes Longitude. Range: 00-59 minutes.
3. Decimal point. Must be there in that position.
4. 2 digits for the fractional part of minutes of Longitude. Range: 00-99
5. 1 character representing the hemisphere. Range: W or E
6. An icon symbol select character (default ">", but see the APRS Specification for details). Required. **THIS CHARACTER IS AUTOMATICALLY CHANGED BY THE ICON MENU.**

NOTE: **Always** use leading zeros for all parts of the Lat/Lon. They **MUST** be there for other APRS stations to correctly interpret the posit.

An example:

Lat: 31 degrees, 43 minutes and 1 second in the Northern Hemisphere.
Lon: 99 degrees, 7 minutes and 56 seconds in the Western Hemisphere.

Here's how it would look as it needs to be entered as a Posit:

=3143.02N/09907.93W>

The fractional part of the Lat (.02) is derived by dividing 1/60 =0.016667 rounded up to .02

The fractional part of the Lon (.93) is derived by dividing 56/60 = 0.93333 rounded down to .93

NOTE: AS SOON AS NMEA DATA IS AVAILABLE, HAMHUD **WILL OVERWRITE** THE MANUALLY ENTERED POSIT.

My Icon: Select Icon For My Station

NOTE: THIS FUNCTION ONLY SELECTS THE ICON FOR THE '='-STYLE POSIT. FOR $GPRMC POSITS, YOU MUST EDIT THE "DESTINATION" ADDRESS IN THE UNPROTO MENU, AND USE A GPSXYZ DESTINATION ADDRESS, SUCH AS GPSMV.

UNPROTO: Set Destination Address and Path

These menu items set the path destination address of the TNC's UNPROTO parameter. In the APRS network, GPS devices (like the HamHUD) can set the icon that will appear on graphical map displays (like WinAPRS^{tm}) by setting an appropriate destination address. See the documentation of these programs for details.

The UNPROTO path controls how packets from the HamHUD will be digipeated. Contact your local APRS club for details on paths.

Selecting the Path menu item brings up the current UNPROTO path. The default path is TRACE4-4, which is an often-used path in WIDEn/TRACEn networks. (Contact your local APRS club for details on paths.) The UNPROTO path can be edited as a string. Double-clicking with the cursor at the right of the path causes it to be saved in the HamHUD's non-volatile memory, and sent out to the TNC.

Note that the HamHUD has no knowledge of the TNC's initial path and icon address; once the HamHUD has sent these items out to the TNC, the current setting is saved in the HamHUD's internal non-volatile memory.

Whenever either the path or the icon/address is changed, BOTH are sent to the TNC in the form of the UNPROTO command:

TNC Path Command Structure::

```
UNPROTO dest V path
```

NOTE: The " V " is a necessary part of the UNPROTO path; HamHUD will not correctly set the TNC's UNPROTO path without it.

Path Shorthand Method:

Since the 4x20 LCD limits string length in editing mode, shortcuts are used to help “squeeze” in more text. The HamHUD employs a handy “shorthand” path method in which WIDE can be represented by a “W” and TRACE can be represented by a “T”. WIDEn-n and TRACEn-n are supported. Once the path has been entered, the HamHUD will internally convert the shorthand form to the normal form before sending it to the TNC.

Entering a path in shorthand method:

Normal Format: WIDE1-1,WIDE3-3

Converts to…

Shorthand Format: W1-1,W3-3

NOTE: **ALL callsigns** (including UNPROTO, MYCALL, message to-calls, etc.) **MUST be entered in UPPERCASE**. This is not HamHUD's fault; the AX.25 protocol specifies that callsigns are UPPERCASE only.

Not all APRS programs accept all destination addresses, however, so take care what is used.

NOTE: If $GPRMC posits are selected (see SmartBeacon menu), you MUST enter a GPS symbol in the destination address, in order for other stations to have an "icon" type for you. For example, for a "car", you would enter:

GPSMV V TRACE3-3 For a TRACEn capable network.

Cntrst/Brite: Setting Display Contrast and Brightness

These menu items control the appearance of the display. Both allow the Whiz Wheel to adjust a number from 0 to 255.

For contrast, smaller numbers result in a darker display. In colder climates, smaller numbers will be necessary to see this display in cold temperatures.

For brightness, larger numbers results in a brighter backlight. Note that the HamHUD will demand more current from the power source the brighter the backlight is set. For minimum power, set the backlight to 0 (off).

When in Beacon Auto (SmartBeacon) mode, the HamHUD will turn the backlight off automatically after the vehicle is stopped for10 minutes. The backlight will return after the vehicle begins moving again. It will also return for any menu operation; it will turn back off after the menu operation is complete.

Line 2: Customizing Your HamHUD's Display

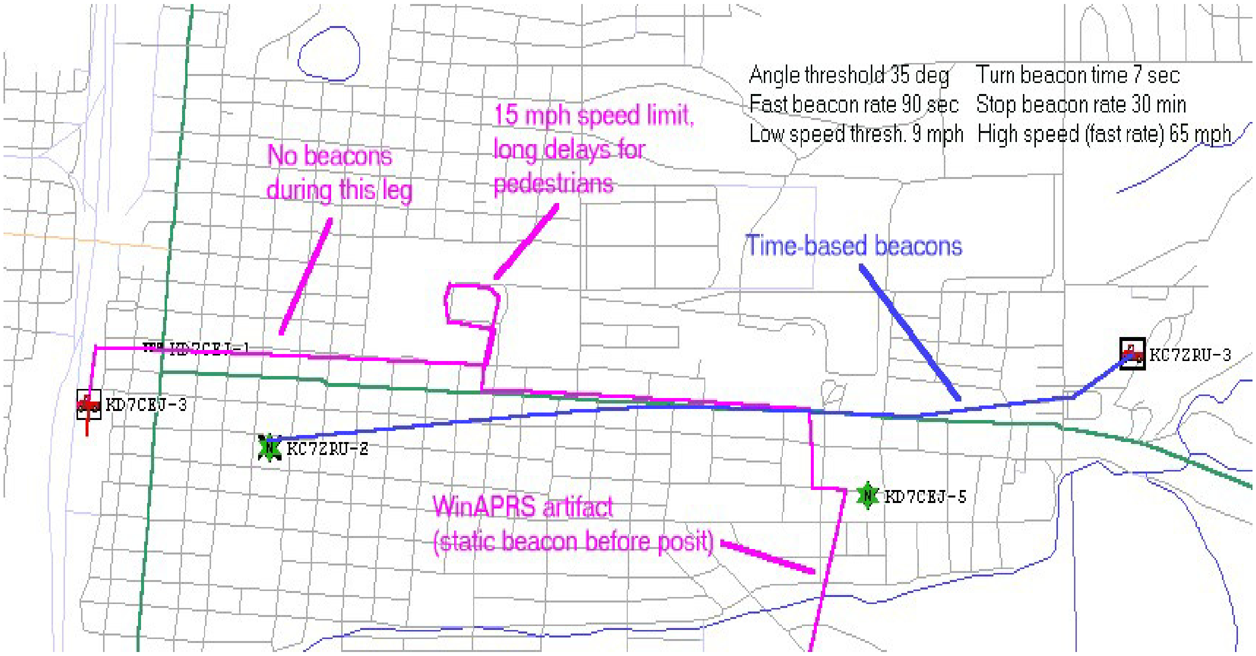
This menu item controls what is displayed in Line 2, the Auxiliary display. Whatever function is on line2 is multiplexed with showing transmitted messages, statues, and queries.

Idle	Shows nothing on line2, except transmitted messages, statues, and queries.
Path	Display 1st 20 characters of the received packets' path. DOES NOT SHOW THE TRANSMITTED PATH.
GPS	Show 1st 20 characters of the $GPRMC sentence from the GPS. Shows GPS time (UTC), and the valid flag (V=unlocked, A=locked).
Posit	Shows your posit, updated as it changes with incoming GPS data.
SmrtBcn	Show and continuously update key SmartBeacon parameters: Beacon_seconds, Seconds_since_beacon, Turn_Threshold,Heading_difference
Hook	Select a station or named waypoint, and continuously update bearing and distance to it. Useful for rendezvous. When the station sends out a new posit, the line2 display is automatically updated.

SmartBeaconing^{tm}

SmartBeaconing is what sets the HamHUD apart from all other devices for mobile APRS. Other devices simply beacon according to a timer, oblivious to the very GPS data that it is sending out. The HamHUD beacons according to speed, heading, distance, and network activity. The HamHUD also "tags" corners, by detecting heading changes from the GPS.

SmartBeaconing is ideal for parades, chase vehicles, sag wagons, SKYWARN operations, etc. The tracks produced by SmartBeaconing in a "good" network environment are both stunningly realistic and economical on bandwidth. They are often visibly superior to "timed" beacons produced by TNCs like the KPC3/+. The following screen shot from WinAPRS™ illustrates this point:



Angle threshold 35 deg Turn beacon time 7 sec
Fast beacon rate 90 sec Stop beacon rate 30 min
Low speed thresh. 9 mph High speed (fast rate) 65 mph

No beacons during this leg
15 mph speed limit, long delays for pedestrians
Time-based beacons
KD7CEJ-1
KD7CEJ-3
KC7ZRU-2
KC7ZRU-3
KD7CEJ-5
WinAPRS artifact (static beacon before posit)

KC7ZRU's path is not readily apparent, while KD7CEJ (with HamHUD on board) traces out a nice trail.

Requirements for SmartBeaconing

For SmartBeaconing to work on the HamHUD, it must have a NMEA 0183 V1.0 or V2.0 stream coming in from a compatible GPS receiver. Only the $GPRMC sentence is currently supported, although the HamHUD will ignore other sentences sent on its GPS input port.

HamHUD cannot SmartBeacon though manual entry of position.

How SmartBeaconing Works

The basic idea of SmartBeaconing, the timed beacon, is in principle little different from the GPS commands of many TNCs. The real difference is that the HamHUD is able to modify its own beacon rate in response to the motion of the vehicle that carries it, thanks to access to the GPS data.

In AUTO mode, SmartBeaconing looks at the $GPRMC sentence and modifies the current beacon rate according to three factors:

1) Distance traveled since last beacon
2) Heading change since last beacon (known as "corner pegging).
3) Network activity.

The algorithms used to implement these are quite simple, but effective. The HamHUD firmware doesn't actually count the distance traveled; it integrates according to the formula d=rt, distance = rate x time. Heading change is calculated by storing the heading at the time of last beacon, and comparing the current heading to it. Network activity control is implemented by simply interleaving display of incoming packets with the beaconing task. With more network activity, the beacons get pushed out in time, effectively slowing the beacon rate with activity.

SmartBeacon Details

To convert distance and heading change to a beacon rate, SmartBeacon uses a three part formula, as shown below in a C-like pseudocode. The braces {} enclose blocks of statements.

```
IF (speed < low_speed)                              // "Stop" threshold
{
        beacon_rate = slow_rate;
}
ELSE  // We're moving; adjust beacon rate to speed, and peg corners
{
// adjust turn threshold according to speed

        turn_threshold = turn_min + turn_slope / mph;

        // Corner pegging

        IF (heading_change_since_beacon > turn_threshold) AND
(secs_since_beacon > turn_time)

{
                CornerPeg = TRUE;
}

    // Adjust beacon rate according to speed

IF (speed high_speed)
        beacon_rate = fast_beacon_rate;
       ELSE
        beacon_rate = fast_beacon_rate * high_speed / speed;
}

if ((secs_since_beacon beacon_rate) OR CornerPeg)
        // ... send beacon
```

So, if the vehicle is stopped, the HamHUD does not evaluate heading changes; it simply adjusts to a fixed (slow) beacon rate. If the vehicle is moving faster than the high_threshold, the HamHUD adjusts to beacon at a fixed (high) rate. In between, the rate decreases linearly with speed.

The basic idea behind the adjustment of turn_threshold is the sort of turns that are made at different speed regimes. For land vehicles, we want more heading-change sensitivity at high speeds than at low speeds. At low speeds, a large turn threshold will do, because almost all turns are sharp ones. At higher speeds, almost all turns are shallow, gradual ones, requiring a small turn threshold to detect. Moreover, the transition between the "low speed" and "high speed" regimes actually takes place (according to our experiments) in the 20-40 MPH range. So, the factors in the equations are geared toward that.

SmrtBeac: SmartBeaconing Parameter Menu

All parameters of the SmartBeacon algorithm are adjustable from the HamHUD's SmrtBeac menu item. The menu item also includes some miscellaneous operating parameters for the HamHUD.

The seven parameters are prompted, one after another. It is only necessary to click the Whiz Wheel if it not desired to adjust a parameter.

To adjust a parameter, turn the Whiz Wheel to the desired value, and click.

After the last parameter is adjusted or dismissed, the menu will exit.

The parameters are prompted in the order they are listed below.

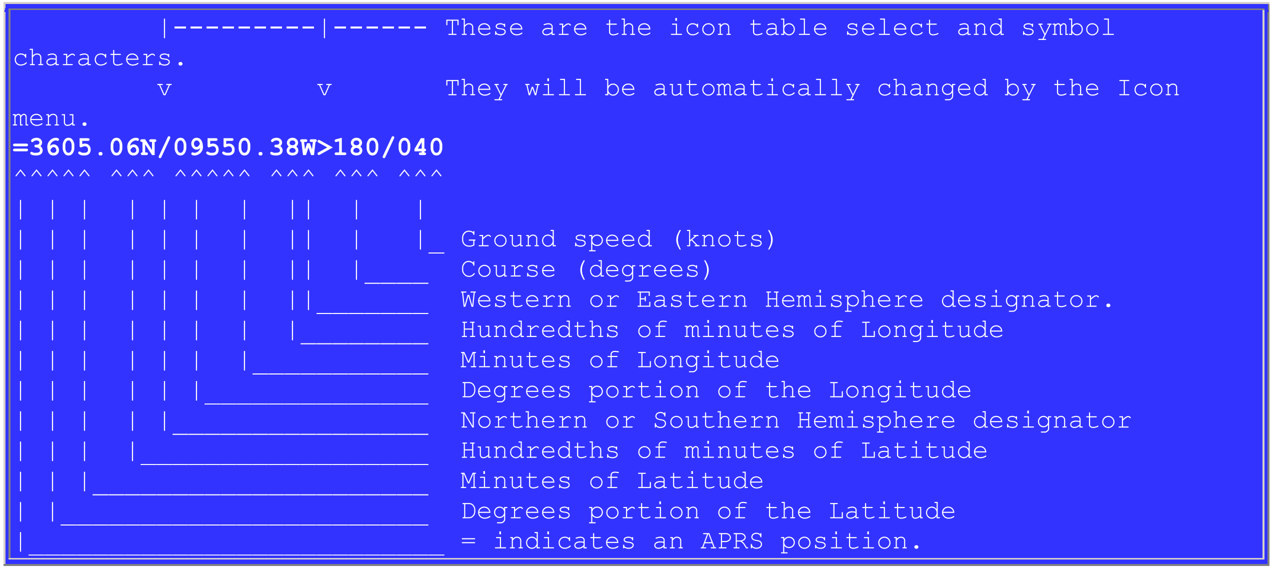
Prompt	Parameter	Description / Effect of Parameter
Turn Ang(deg)	turn_Angle	Minimum angle for dynamically-adjusted turn-beacon threshold. (DEG).
TSlp (10deg/u)	turn_Slope	Multiplier for dynamically-adjusted turn beacon threshold. (10*DEG-MPH)
Turn Dly(sec)	turn_Time	Minimum time between any other beacon and a turn beacon (SEC)
FastRate(sec)	beacon_Fast	Fast beacon rate (SEC)
SlowRate(mins)	beacon_Stop	Stop beacon rate (MIN)
LoSpeed(units)	Low_speed	Speed (MPH) below which SmartBeaconing is disabled, and we beacon at the beacon_stop rate
HiSpeed(units)	High_speed	Speed (MPH) above which HamHUD beacons at the beacon_fast rate
Status (mins)	Status_mins	Time (MIN) between transmits of the currently-selected status string.
0=Posit >0=RMC	Posit_Format	Setting 0 selects the efficient '=' posit format. Setting any other number selects the $GPRMC sentence format. NOTE: YOU must manually select the icon with the destination address (e.g., GPSMV) when using $GPRMC format.
0=US >0=Metric	Units	Setting 0 selects the display units to United States (Imperial). Speed is in statue miles per hour, distance is in statute miles, and temperature is in degrees Fahrenheit. Setting any other number selects SI (metric) units. Speed is in kilometers per hour, distance is in kilometers, and temperature is in degrees Celsius.

Posit_Format

HamHUD sends the efficient '=' position ("posit") format by default. Instead of sending 70-plus characters on every sentence, like $GPRMC does, '=' format only requires 27 characters. It gives position accurate to 1/100th of a mile, speed accurate to 1 knot, and course in degrees. The only thing better about $GPRMC is the position accuracy, which can be as accurate as a few feet with DGPS.

Please do not use the $GPRMC format unless you absolutely NEED the position accuracy. The extra network traffic is **NOT WORTH IT** in most cases. When you select the NMEA ($GPRMC) format, you will have to set the icon yourself, by changing the APRHH2 V TRACE3-3 in the UNPROTO menu to something like GPSMV V TRACE3-3. **YOU CANNOT USE THE BUILT-IN ICON SELECTOR with NMEA format**. Also, the '=' format (as edited in the Posit menu) will still be sent when there is no GPS data.

The figure below illustrates the position format.



```
         |---------|------ These are the icon table select and symbol
characters.
         v         v       They will be automatically changed by the Icon
menu.
=3605.06N/09550.38W>180/040
^^^^^ ^^^ ^^^^^ ^^^ ^^^ ^^^
| | |  | | |  |  ||  |   |
| | |  | | |  |  ||  |   |_ Ground speed (knots)
| | |  | | |  |  ||  |____  Course (degrees)
| | |  | | |  |  ||_______  Western or Eastern Hemisphere designator.
| | |  | | |  |  |________  Hundredths of minutes of Longitude
| | |  | | |  |___________  Minutes of Longitude
| | |  | | |______________  Degrees portion of the Longitude
| | |  | |________________  Northern or Southern Hemisphere designator
| | |  |__________________  Hundredths of minutes of Latitude
| | |_____________________  Minutes of Latitude
| |_______________________  Degrees portion of the Latitude
|__________________________ = indicates an APRS position.
```

TNC Cmd: Sending a Command to the TNC

This menu item allows a command to be sent directly to the TNC. The command is limited to 20 characters long, and there are six different editable commands.

As commands vary between versions of TNC even with the same manufacturer, the user is strongly urged to consult the TNC manual before using this menu item. **It is possible to issue commands that will prevent the HamHUD from communicating with the TNC.** Use EXTREME caution.

TNCCmd cannot display the response sent back from the TNC. Therefore, it is not useful for TNC commands that return a response.

Upgrading the HamHUD Firmware and Initializing the FRAM (Windows Loader):

- The older DOS firmware loader has been replaced with a Windows based version that allows the user to have graphical interface that makes loading new firmware a snap!

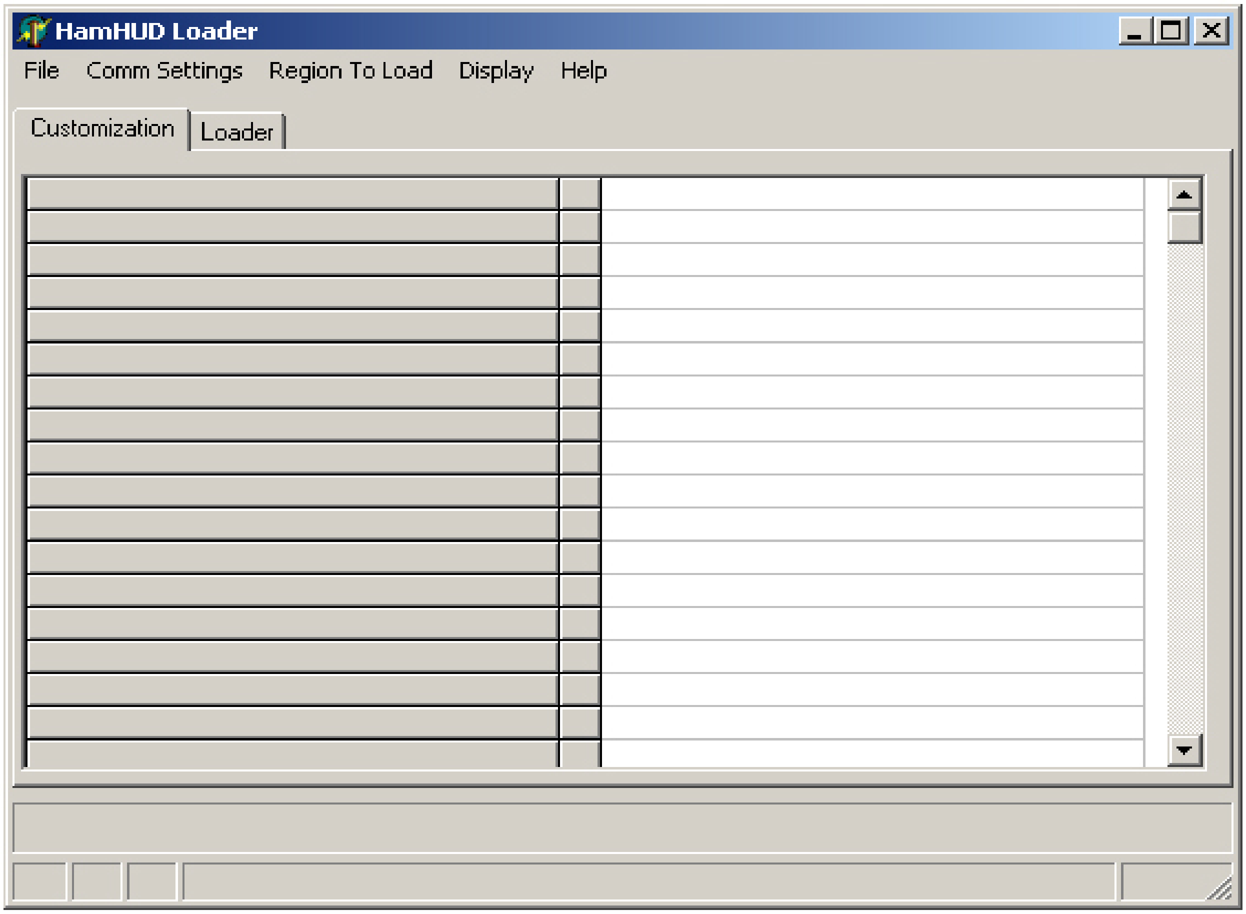
- In order to use this loader, your PIC MUST be initially burned with a normal PIC programmer such as a Warp13 or Olimex Picstart with a stable HamHUD II firmware version. This is required to have the bootloader in place to talk to the Windows loader.

- **Important!** There are many flaky problems with loading through USB/Serial adapters and/or splitters. A hardware serial port is always recommended to insure a proper load and to help avoid disabling your PIC. Failure to load the PIC will result in the requirement to have the PIC re-burned with one of the above mentioned programmers.

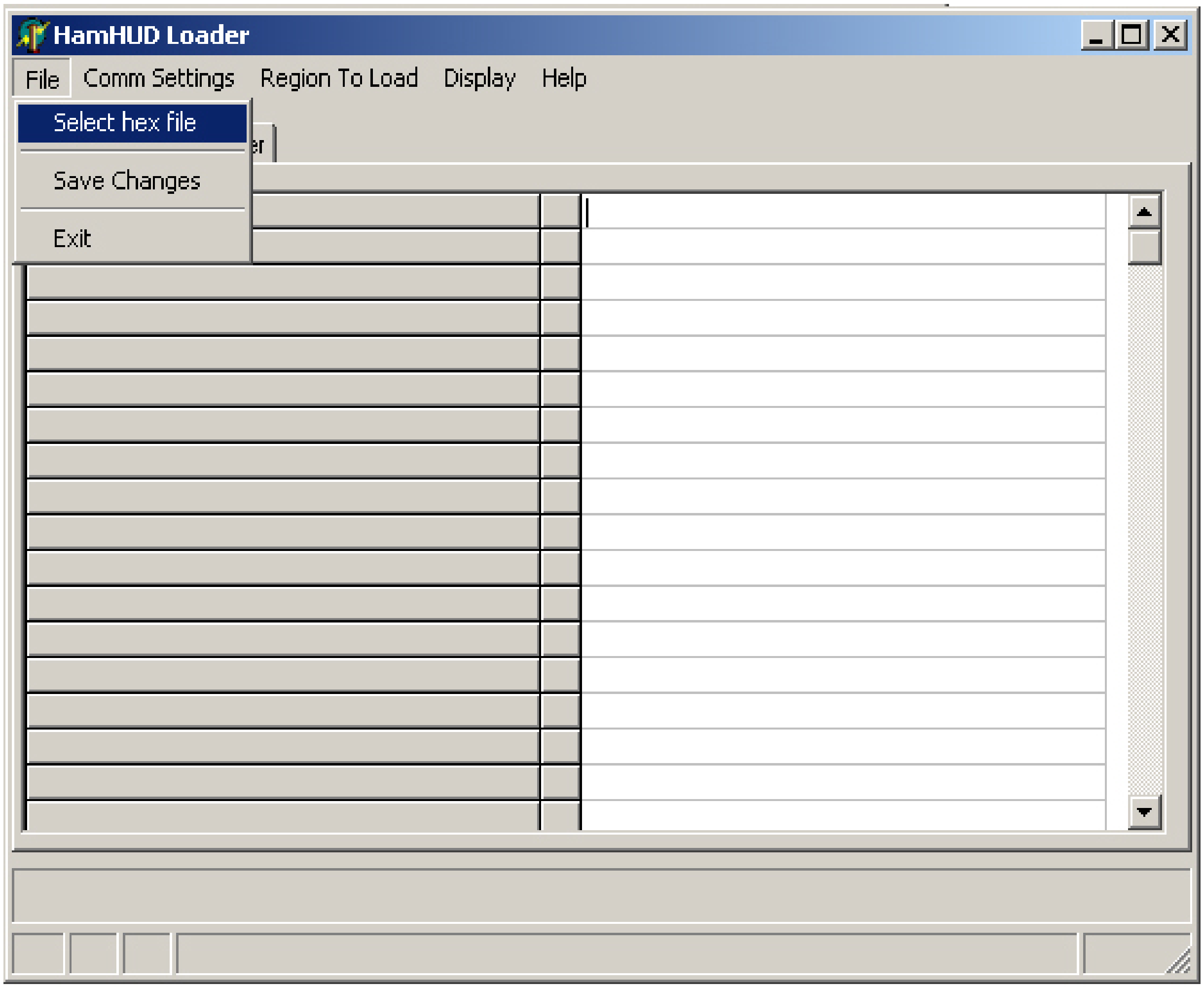
Instructions for loading firmware:

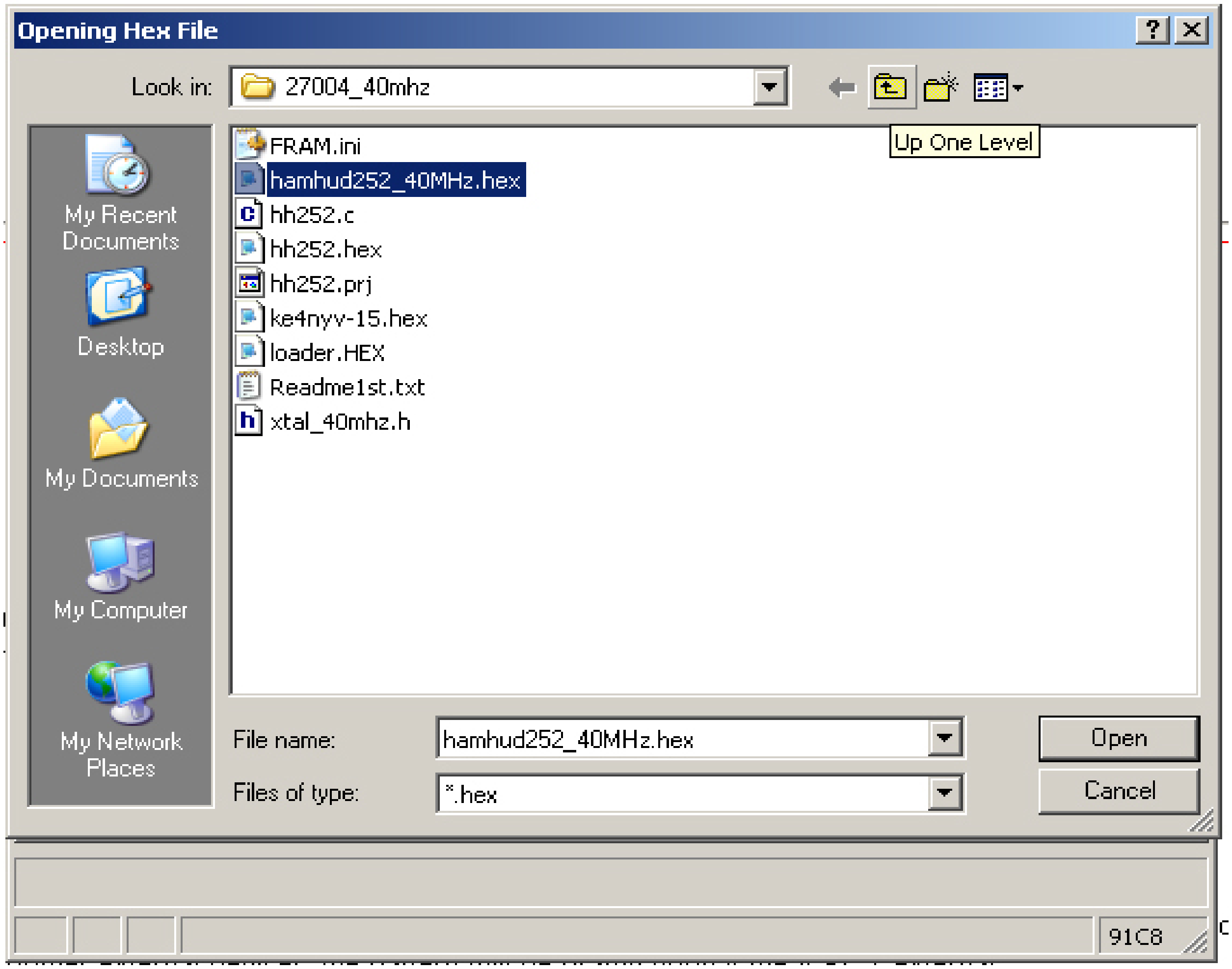
Step 1: Start the HamHUD II in LOAD MODE. Make sure the HamHUD II Is powered off. Hold in the WhizWheel button and turn the HamHUD II back on while holding in the button. Wait for "LOAD" to appear on the top line. You may now release the WhizWheel button and you are ready to load firmware.

Step 2: Open the Loader



Step 3: Select hex file from the location you downloaded the firmware to.





Opening Hex File

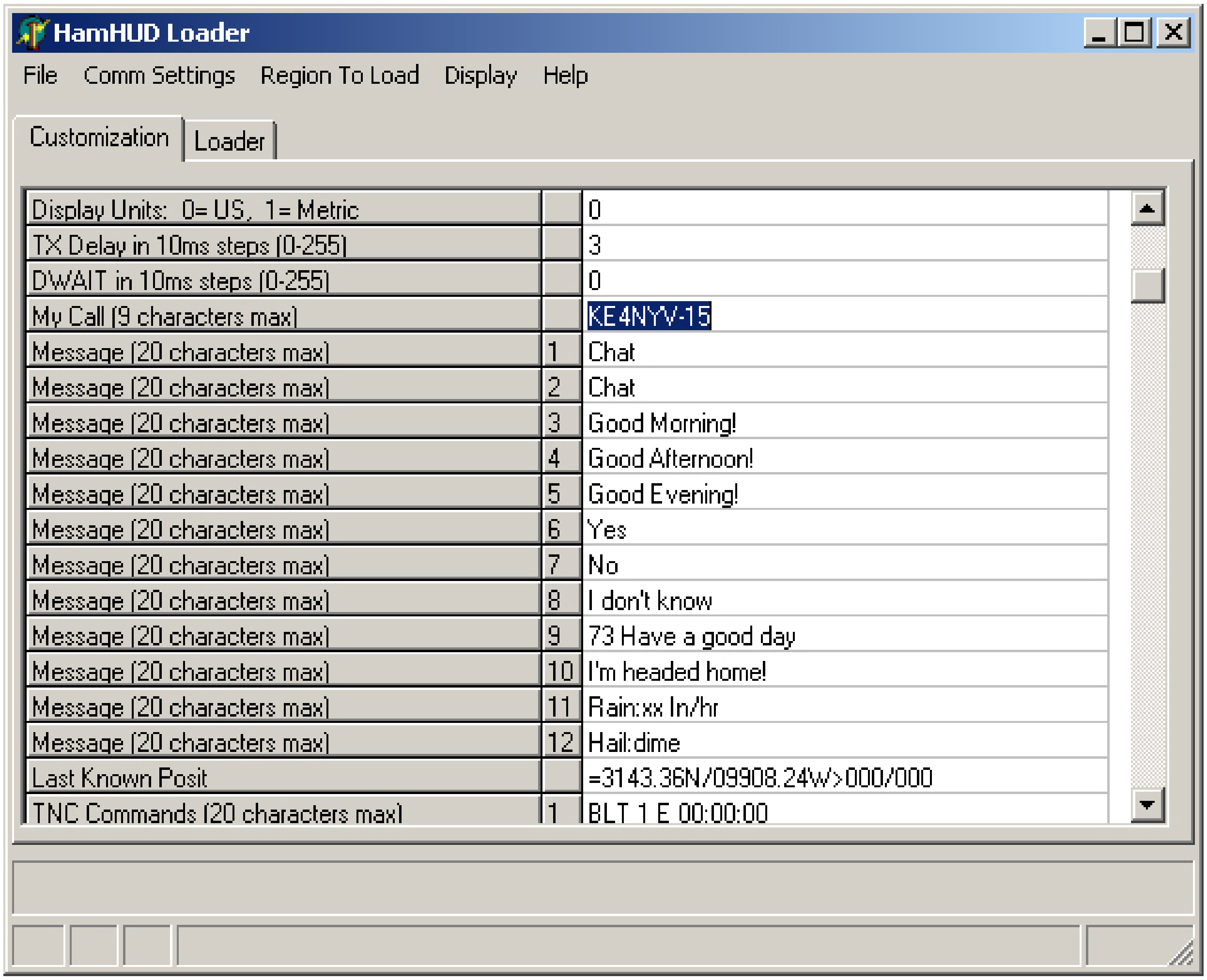
Look in: 27004_40mhz

FRAM.ini
hamhud252_40MHz.hex
hh252.c
hh252.hex
hh252.prj
ke4nyv-15.hex
loader.HEX
Readme1st.txt
xtal_40mhz.h

File name: hamhud252_40MHz.hex

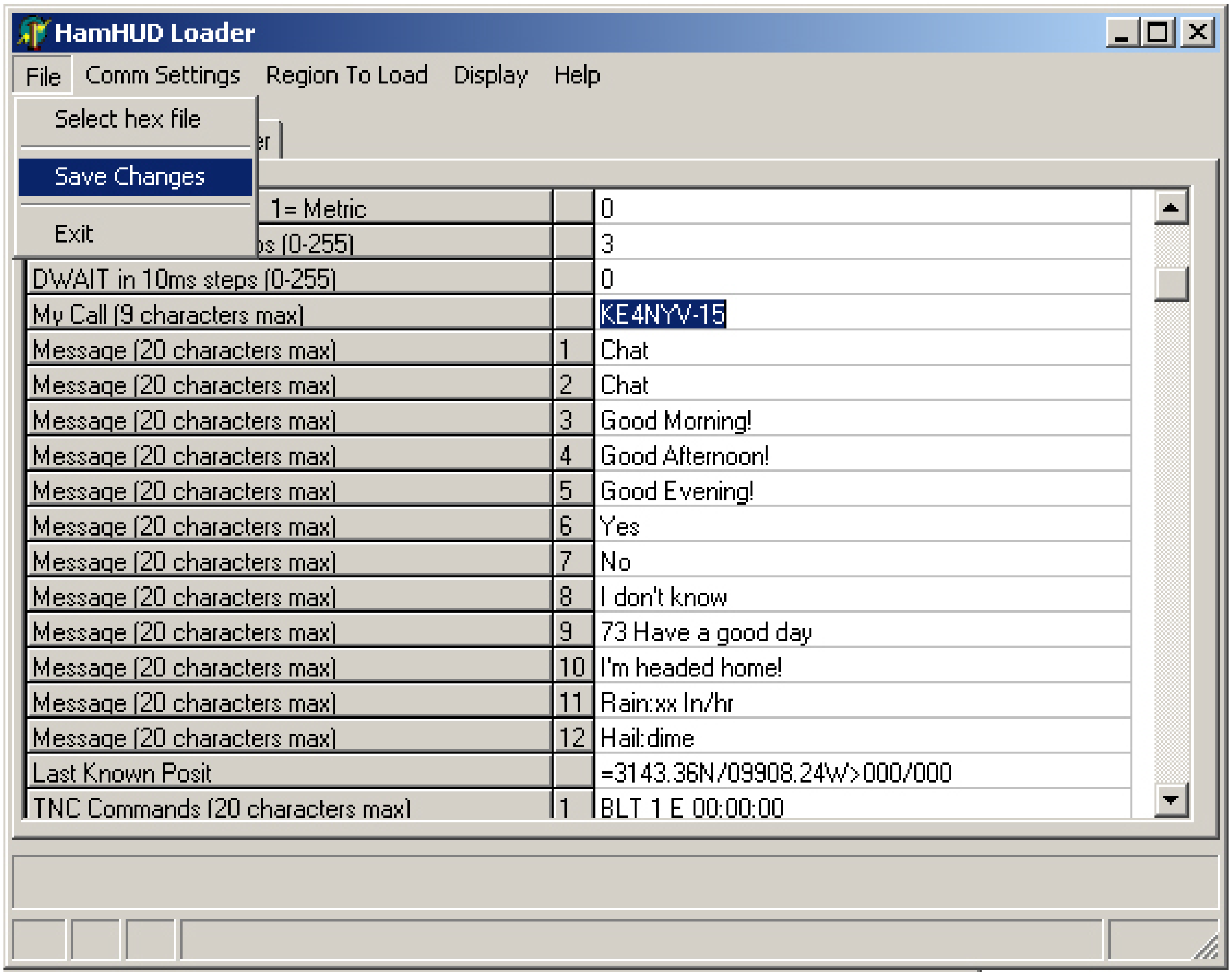
Files of type: *.hex

Step 4: Make any changed to the FRAM data at this time. Callsign is the only required change. Make this callsign match what you put in the TNC.

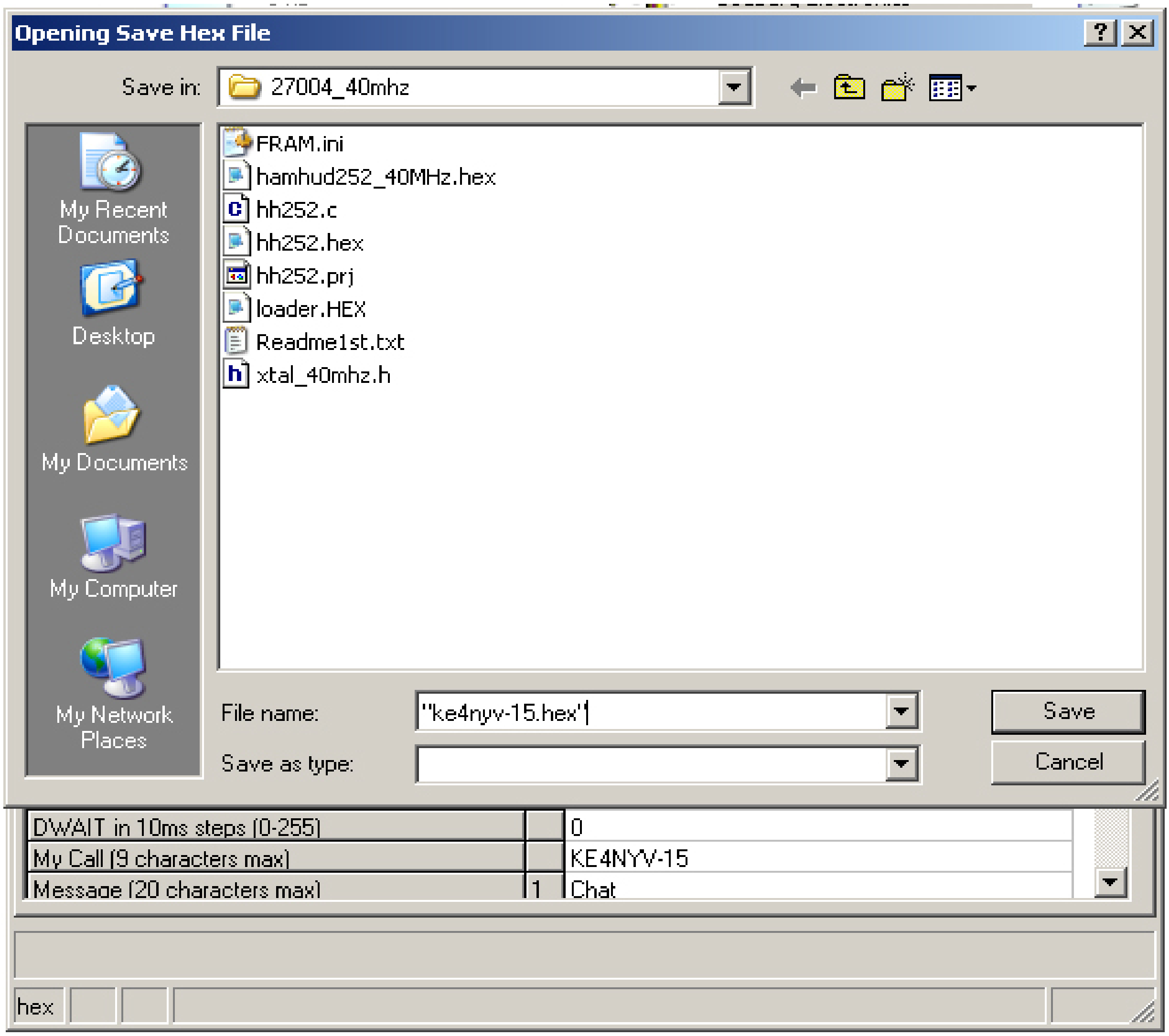


Display Units: 0= US, 1= Metric		0
TX Delay in 10ms steps (0-255)		3
DWAIT in 10ms steps (0-255)		0
My Call (9 characters max)		KE4NYV-15
Message (20 characters max)	1	Chat
Message (20 characters max)	2	Chat
Message (20 characters max)	3	Good Morning!
Message (20 characters max)	4	Good Afternoon!
Message (20 characters max)	5	Good Evening!
Message (20 characters max)	6	Yes
Message (20 characters max)	7	No
Message (20 characters max)	8	I don't know
Message (20 characters max)	9	73 Have a good day
Message (20 characters max)	10	I'm headed home!
Message (20 characters max)	11	Rain:xx In/hr
Message (20 characters max)	12	Hail:dime
Last Known Posit		=3143.36N/09908.24W>000/000
TNC Commands (20 characters max)	1	BLT 1 E 00:00:00

Step 5: Save the FRAM changes to a NEW file. We suggest your callsign. Use the quotes “ “ around the filename and the .hex



1= Metric		0
ps (0-255)		3
DWAIT in 10ms steps (0-255)		0
My Call (9 characters max)		KE4NYV-15
Message (20 characters max)	1	Chat
Message (20 characters max)	2	Chat
Message (20 characters max)	3	Good Morning!
Message (20 characters max)	4	Good Afternoon!
Message (20 characters max)	5	Good Evening!
Message (20 characters max)	6	Yes
Message (20 characters max)	7	No
Message (20 characters max)	8	I don't know
Message (20 characters max)	9	73 Have a good day
Message (20 characters max)	10	I'm headed home!
Message (20 characters max)	11	Rain:xx In/hr
Message (20 characters max)	12	Hail:dime
Last Known Posit		=3143.36N/09908.24W>000/000
TNC Commands (20 characters max)	1	BLT 1 E 00:00:00



Opening Save Hex File

Save in: 27004_40mhz

FRAM.ini
hamhud252_40MHz.hex
hh252.c
hh252.hex
hh252.prj
loader.HEX
Readme1st.txt
xtal_40mhz.h

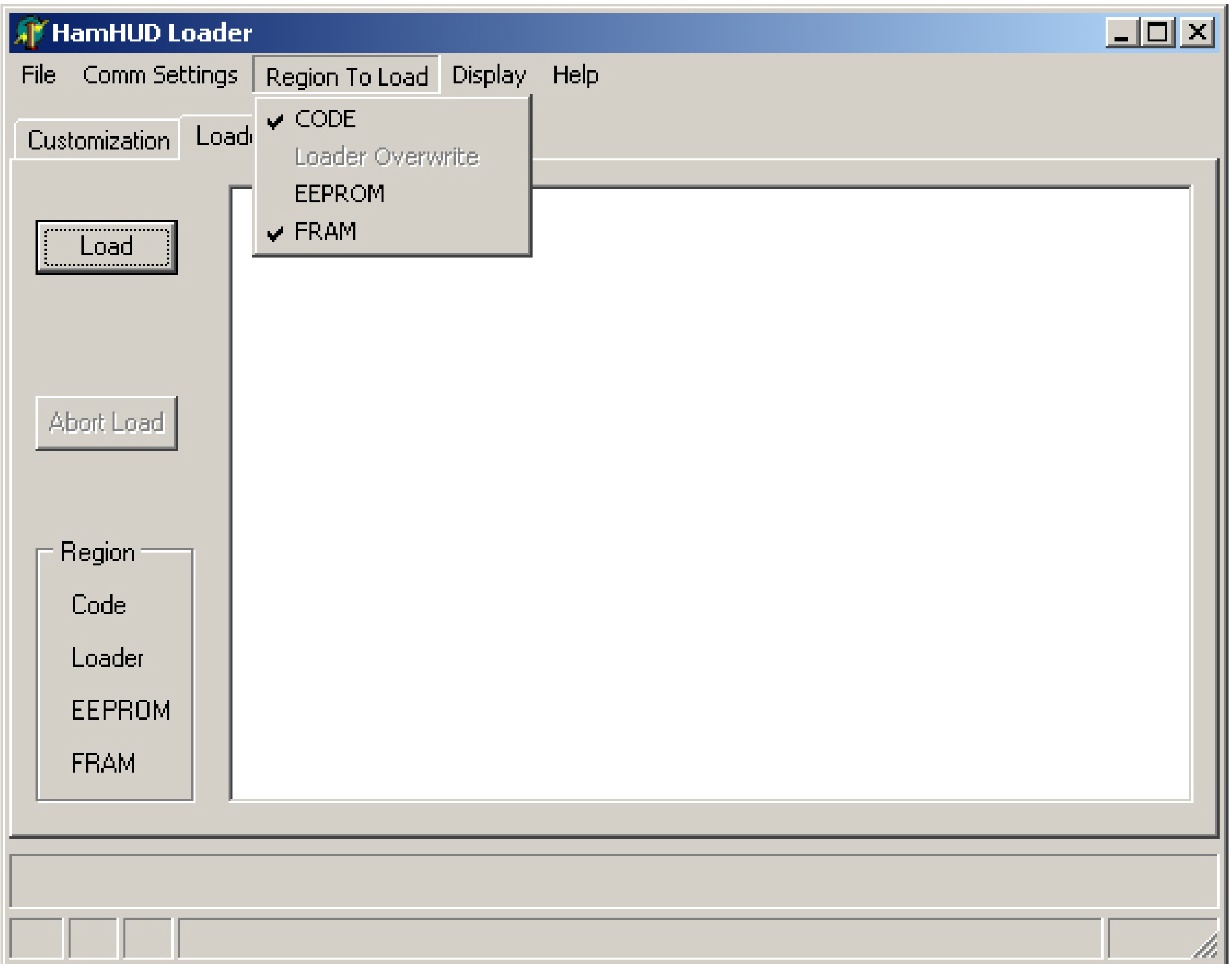
File name: "ke4nyv-15.hex"

Save as type:

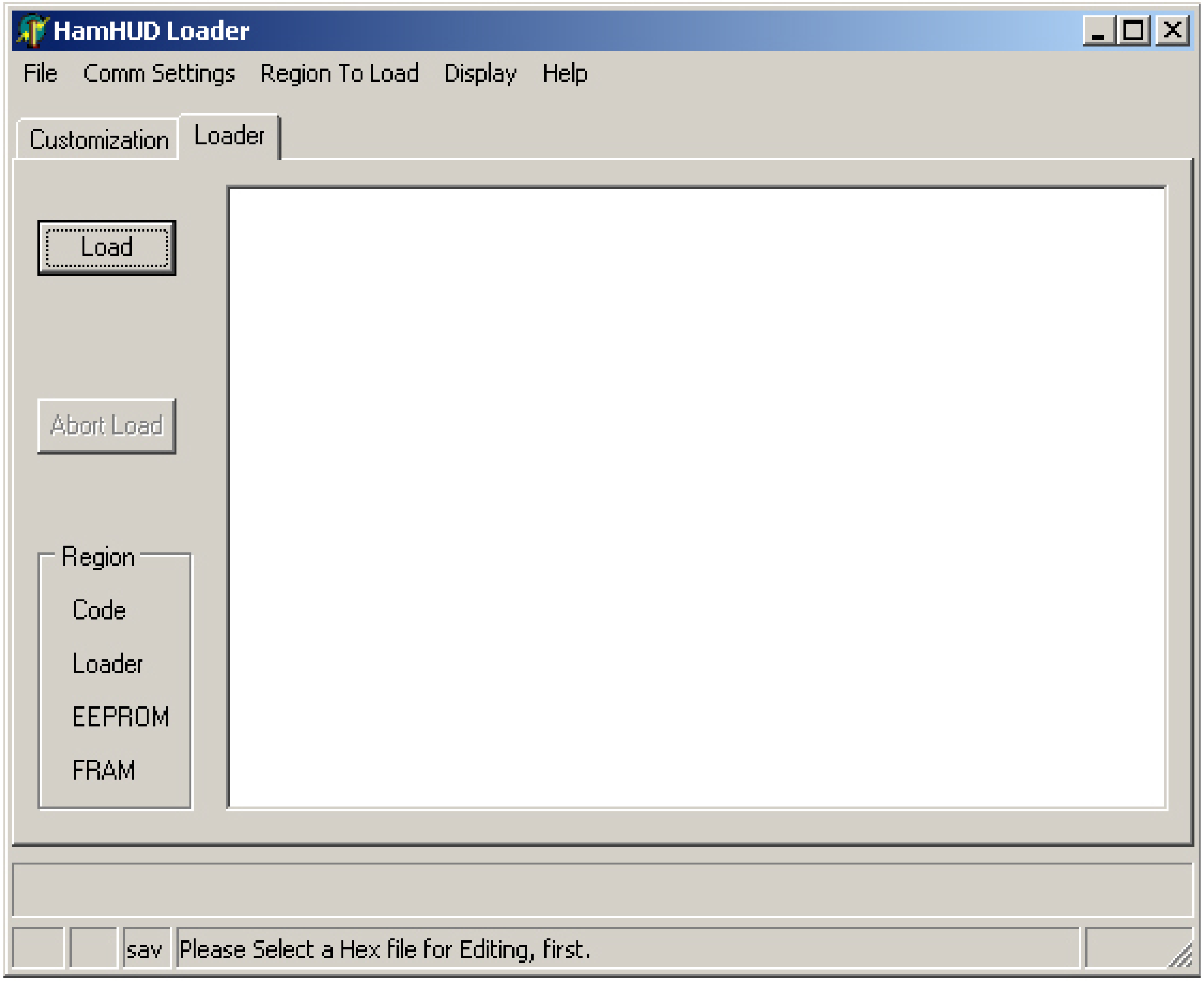
DWAIT in 10ms steps (0-255)		0
My Call (9 characters max)		KE4NYV-15
Message (20 characters max)	1	Chat

hex

Step 6: Select the reqions you would like to load. If this is a first-time load, you need to select both CODE and FRAM. Future firmware upgrades can omit FRAM to retain settings.



Step 7: Click the Load button. Allow the HamHUD to go all the way through the load. When it's done, the loader will tell you to reboot the HamHUD to let the new firmware and FRAM values load up.



Please Select a Hex file for Editing, first.

APPENDIX A: KPC3+ Jumpers

[As far as I know, the CD light only works on the KPC3+.]

This jumper was made from a 3-pin header socket with the center contact pulled out. Or, you can simply solder a wire between them.

Power

J6 1[C2] Routes Vin to J7(C)
J7 [1C3] All 3 together, routes Vin to DB9(7) and DB25(13)

NOTE: Many KPC-3+ users install the jumpers at J1 and J2 to allow for powering the TNC either from the internal battery or the external power input, whichever happens to be present. However, there is an undesirable side effect if the special J7 jumpering is used to power external devices: the battery will be drawn upon if the KPC's external power is interrupted. This can drain it completely, and rather quickly in fact.

"CD" light (via the KPC3+'s CTRL A FET)

J10 C[12] Special jumper (3 pins wide but avoids center pin; a wire can also be used), which routes the Fet to DB25[11], a.k.a. AD1 CTRL A MUTE n TNC command is required, n can be 1 to 3. The MUTE parameter is available only in the KPC-3+ version 8.3.

[de Steve VE3XF] Also, a 2 pin jumper will connect between the correct pins on J8 and J10 bringing the CD line out to pin 18 on the DB25. This might be the simplest thing to do and it leaves the other two pins on each connector untouched.

Alternate "CD", using the CTRL B FET

J8 C[12] Same jumper that you would have used on J10. CTRL B MUTE n commande would be required.

If you wish to use the Analog Input #1 (AN1) or Analog Input #2 (AN2), you would have to connect to J10(C) of J8(C), avoiding the outer 2 pins.

If you wish to bring out a "CD" signal on any other rev of the KPC-3+, or on any other TNC model, you would have to connect a jumper wire to the "cold" (ground) side of the front panel "RCV" LED.

APPENDIX B: HamHUD to TNC/GPS cables and Pinouts

- From the first generation HamHUD II's until mid 2006, it was always up to the end user to decide on the connector interface and method they would use to connect their HamHUD to their TNC, GPS and Power. While there are still many of these HamHUD II's out there in operation, there is a new standard being use by the HamHUD community with the RevE kits. In this section, we will attempt to give some general serial pinouts useful for the older generation HamHUD as well as the pinouts for the newer RevE kit models.

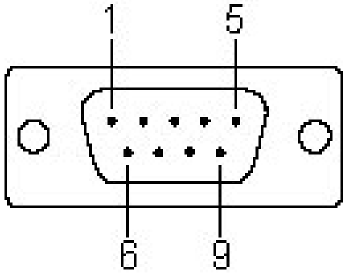
Standard Configuration:

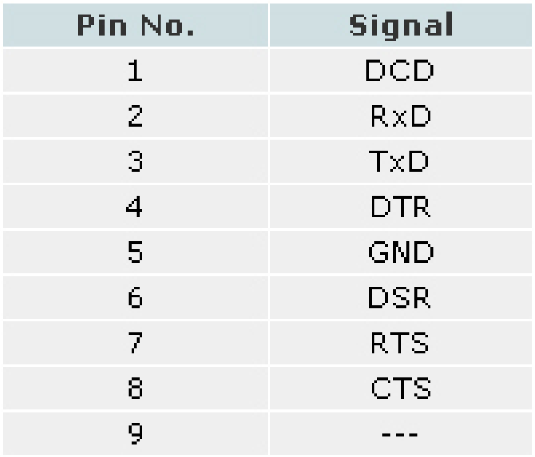
Radio → TNC → HamHUD ← GPS

In the older configuration, there are a host of possible connector options for both the GPS and TNC to the HamHUD. The most common are D-Sub DB9 and/or DB25 connectors.

- When wiring the HamHUD, you have to think of it as a replacement for a full size computer with two normal serial ports. One to handle the TNC comms and one to handle the GPS comms. Wire the ports to accept the device's cables directly without any adapters or crossover/null adapters. (A null configuration will only be required when programming.)
- Here are some common serial port pinouts:

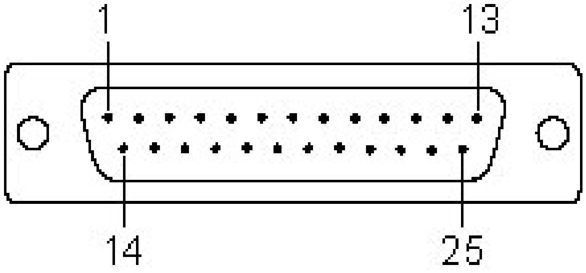
D-Sub 9 Pin (Male)

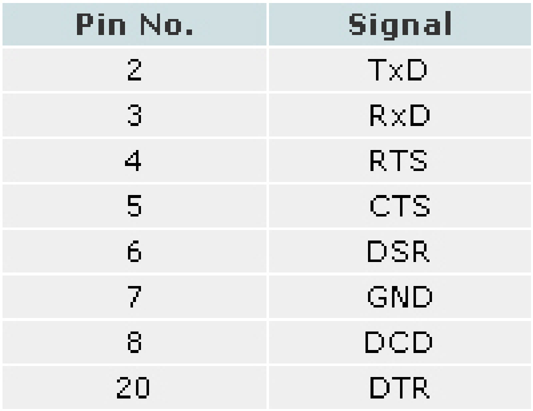




Pin No.	Signal
1	DCD
2	RxD
3	TxD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	---

D-Sub 25 Pin (Male)





Pin No.	Signal
2	TxD
3	RxD
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
20	DTR

APPENDIX C: USING A TNC STATIC BEACON

To increase the reliability of the TNC/HamHUD system, it may be desirable to initiate a long-duration static beacon in the TNC. The advantage of this is that if the HamHUD is somehow turned off or taken out of service (perhaps by a car thief!), the TNC will continue to beacon.

To do this, a TNC is required that can support a second serial port for the GPS, such as the Kantronics KPC3+. For the KPC3+, the following settings are recommended:

```
GPSTIME VALID RMC
GPSHEAD 1 $GPRMC
LTP 1 GPSMV VIA TRACE3-3
BLT 1 E 00:20:00
```

These commands set up a periodic ("static"). HamHUD can control this beacon by sending commands via the TNCCmd menu. The "GPSMV" should be replaced with whatever icon the user desires.

Since the static beacons will come from the same callsign as the automatic ones, they will be recognized by the HamHUD. SmartBeacon will reschedule the automatic beacons accordingly.

APPENDIX D: KNOWN BUGS

- When Line2 = SB display, and Beacon is set to any fixed value, the "angle since last beacon" field of line2 is fixed at a value of zero, irrespective of any actual heading variations.
- HamHUD does not process the "old" object format that WinAPRS generates.
- Doesn't show "MY" path on line2 on xmit
- Status interval = 0 IS BAD! Need to make interval=0 mean don't transmit status.
- Lowercase callsigns are TOXIC