

# Host Interface Spec for the ARDOP TNC (preliminary)

Rick Muething, KN6KB

Revised: Sept 12, 2016

(This should agree with ARDOP\_Win Beta release 0.6.3)

Please send questions, comments or suggestions to

<mailto:rmuething@cfl.rr.com>

## 1.0 Scope

This is the Host interface specification for those wishing to use an ARDOP compatible TNC (virtual Software TNC or hardware/firmware TNC) in host applications. The ARDOP protocol specification document should be used in conjunction with this host specification.

## 2.0 Protocol Name and Purpose

The protocol name is ARDOP (Amateur Radio Digital Open Protocol). This document details how a host program (e.g. a radio client or server) can interface to an ARDOP modem. The details in this document cover the ARDOP Windows virtual TNC but should also be pretty close for other modems which could be either a pure software implementation (a virtual TNC) or a physical hardware/DSP firmware implementation that implements the ARDOP protocol

## 3.0 Initialization

The host program would normally use some initialization file (e.g. "ARDOP TNC.ini") that resides with the host program. That file should contain the minimum information to allow the host program to start the ARDOP TNC automatically. This includes:

TCP IP Control Port number used for both command and data (one port) and the Address for that port (e.g. 127.0.0.1 if on the same computer)

Or

Serial COM Port and baud rate if using Serial interface

Or

Bluetooth interface details for pairing if the Bluetooth interface to the TNC is used.

If the ARDOP TNC is a virtual TNC (software + sound card implementation) it can be launched from a command line with the above necessary parameters which are sufficient to initialize the Host <> Virtual TNC link.

e.g. for TCPIP connection from Host to Virtual TNC:

```
Shell("ARDOP_Win.exe TCPIP 8515 127.0.0.1")
```

Would start the ARDOP\_Win virtual TNC using a TCPIP port listening on port 8515 of address 127.0.0.1

e.g. for Serial (RS232 or USB) connection from Host to Virtual TNC:

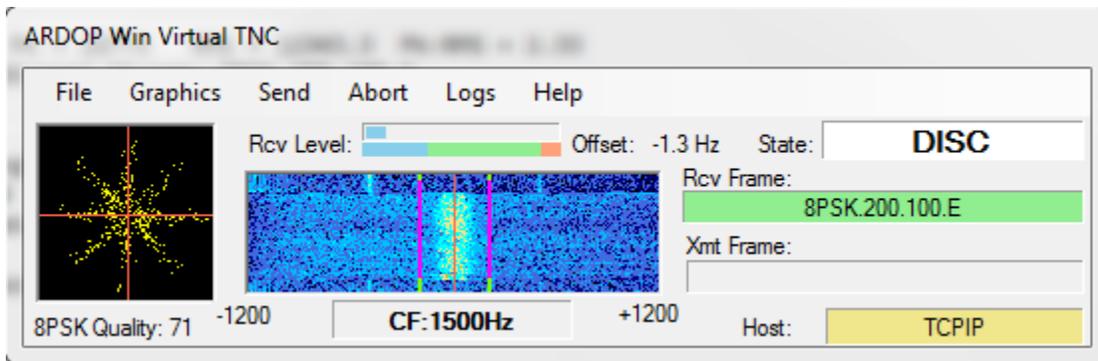
Shell("ARDOP\_Win.exe SERIAL COM5 38400")  
Would start the ARDOP\_Win virtual TNC using a serial port listening on COM5  
baud rate of 38400.

e.g. for BlueTooth connection from Host to Virtual TNC: (*preliminary*)  
Shell("ARDOP\_Win.exe BLUETOOTH ARDOPWINTNC")  
Would start the ARDOP\_Win virtual TNC using a BlueToothl port pairing to  
"ARDOPWINTNC".

All remaining property values and setting may be supplied using the commands  
list or in some cases may use a separate ini file for ARDOP virtual TNC

#### 4.0 ARDOP TNC User Interface

The host program normally would include a parameter setup menu for the ARDOP TNC. The ARDOP Win TNC's setup menu (and resulting .ini file) will however allow setup of all parameters and host interface options and this is often helpful during initial testing and TNC/Host integration. The ARDOP Win TNC also includes a "virtual panel" which may be made visible or hidden by the host application. It is intended to provide the user with entertainment, information and real time status of the ARDOP TNC and status of any existing connections but is *not required* for operation. The following is a screen capture of the ARDOP Win virtual TNC. The virtual TNC interface is optional and may be eliminated for example if a hardware/firmware DSP implementation provides any needed status information.



**Fig 4-1 ARDOP Win TNC User Interface "Virtual Panel"**

The above panel shows the reception of a single carrier (200 Hz bandwidth) 8PSK, 100 baud frame with a successful decode and a decoded quality value (0-100) of 71. The channel was multipath poor at 0 dB S/N. The virtual TNC is using a TCPIP Host<> TNC interface. The Rcv and Xmt frame labels indicate the frame type being received or transmitted. State information is also shown which matches the state diagrams in the ARDOP protocol specification. The Green or Magenta tuning lines in the waterfall indicate the detection of a signal with the TNC's busy detector (Green = NOT BUSY, Magenta = BUSY). More details on this specific interface are described in the Help file for the ARDOP\_Win TNC.

## 5.0 Command Interface

The command interface is via ASCII commands sent to the TCPIP address/port, the serial port or to the Bluetooth paired device identified in the host .ini file. It is designed for a computerized host interface (not keyboard commands as in 1980's type Packet TNCs). For robustness all serial or Bluetooth commands and data are protected by a 16 bit CRC. TCPIP commands are not CRC protected as the TCPIP protocol itself has built in CRC and auto repeat if necessary. All commands are plain (readable) text using 7 bit ASCII character encoding. Commands sent from the host begin with "C:" and terminate with a <Cr> followed by a 2 byte CRC check for non TCPIP interfaces. No command name or parameter may include the strings "C:", "c:", or the <Cr> character. Commands that do not include parameters will cause an "echo back" of the current parameters as described below. Upon initial connection to the TCP port at the TCP Address, the COM port or the Bluetooth paired device in the .ini file the host connect request will be accepted and the following response issued: "c:RDY<Cr>+2 byte CRC". Commands or replies from the TNC always begin with "c:" (lower case c) and end with a "<Cr>+2 byte CRC". After *completing* execution of a command the virtual TNC will respond with: "c:RDY<Cr>" indicating it is available for the next command. Commands *must* be sequenced using this RDY response. Note some commands may take a few seconds to execute.

If a command or data from the host is received and has a CRC failure the Virtual TNC will reply with "c:CRCAFAULT<CR>+ 2byte CRC". The host should then repeat the command or data transfer.

If a command from the host is received with improper syntax or parameter values the Virtual TNC will reply with "c:FAULT <fault description + Echo back of command><CR>+2 byte CRC". Followed immediately with a RDY command. A CRC fault (very unlikely with a TCPIP connection) should prompt the Host to repeat the command.

If debug logging is enabled and Enable Command Trace is enabled each command will be logged as received and each response will be logged to the debug log as sent. This is helpful in initial debugging the host application interface. Debug logging carries an accurate (1 ms resolution) time tag.

If the Host decodes a CRC Failure on any command or data it must reply with a CRCAFAULT command (in lieu of the normal RDY command which will prompt the TNC into resending the command. The TNC will repeat the command up to 2 times before declaring an interface fault.

The following is the current list of commands with Explanation. What is between the "<>" are the command options separated by "|" or the response given in the reply to the command.

Items marked in **blue** text are anticipated but not yet implemented.

**ABORT** Immediately aborts an ARQ Connection (dirty disconnect) or a FEC Send session.

**ARQBW <200MAX|500MAX|1000MAX|2000MAX|200FORCED|500FORCED|1000FORCED|2000FORCED>**

Set/gets the bandwidth for ARQ mode. This sets the maximum negotiated bandwidth or sets the forced bandwidth to a specific value. Attempting to change bandwidth while a connection is in process will generate a FAULT. If no parameter is given will return the current bandwidth setting. This bandwidth setting applies to all call signs used (MYCALL plus optional call signs MYAUX)

**ARQTIMEOUT<30-600>** Set/get the ARQ Timeout in seconds. If no data has flowed in the channel in ARQTimeout seconds the link is declared dead. A DISC command is sent and a reset to the DISC state is initiated. If either end of the ARQ session hits it's ARQTIMEOUT without data flow the link will automatically be terminated.

**ARQCALL<Target Callsign Repeat Count>** Target Call sign must be a legitimate call sign syntax or "CQ" Repeat count must be 2 to 15 e.g. ARQCALL W1AW 5

**AUTOBREAK <False|True>**

Disables/enables automatic link turnover (BREAK) by IRS when IRS has outbound data pending. Default = True.

**BREAK**

Initiates a BREAK (link turnover request to the ISS) if in IRS state otherwise it has no effect. Forces ISS to clear its outbound queue and acknowledge with ACK. Normally not required if AUTOBREAK is enabled. May take up to 6 seconds to respond.

**BUFFER**

Gets the current outbound data buffer size in bytes. BUFFER is also sent asynchronously whenever there is a change in the outbound buffer size.

**BUSYLOCK <False|True> (not currently implemented)**

Disable/Enable Busy channel blocking. Busy lock will block a connection request unless there have been  $T_{\text{quiet}}$  ms of non busy status preceding the connect request. (See appendix B) Default = False

**CAPTURE <device name>**

Sets desired sound card capture device If no device name will reply with the current assigned capture device.

**CAPTUREDEVICES**

Returns a comma delimited list of all currently installed capture devices.

**CLOSE**

Provides an orderly shutdown of all connections, release of all sound card resources and closes the Virtual TNC Program or hardware.

**CMDTRACE**<True|False> Get/Set Command Trace flag to log all commands to from the TNC to the ARDOP\_Win TNC debug log.

**CODEC** <False|True>

Start the Codec with True, Stop with False. No parameter will return the Codec state.

**CWID** <False|True>

Disable/Enable the CWID option. CWID is optionally sent at the end of each ID frame.

**DATASEND** <0>

If sent with the parameter 0 (zero) it will clear the TNC's data to send Queue. Cleared data can be recovered using the RESTORBUFFER command. If sent without a parameter will return the current number of data to send bytes queued.

**DEBUGLOG** <False|True>

Enable/disable the debug log.

**DISCONNECT**

Initiates a normal disconnect cycle for an ARQ connection. If not connected command is ignored.

**DISPLAY**<Frequency in KHz>

Sets the Dial frequency display of the Waterfall or Spectrum display. If sent without parameters will return the current Dial frequency display. If > 100000 Display will read in MHz.

**DRIVELEVEL** <0-100>

Set Drive level. Default = 100 (max)

**FECID**<False|True>

Disable/Enable ID (with optional grid square) at start of FEC transmissions

**FECMODE**<8FSK.200.25|4FSK.200.50S|4FSK.200.50,4PSK.200.100S|4PSK.200.100|8PSK.200.100|16QAM.200.100|16FSK.500.25S|16FSK.500.25|4FSK.500.100S|4FSK.500.100|4PSK.500.100|8PSK.500.100|16QAM.500.100|4FSK.1000.100|4PSK.1000.100|8PSK.1000.100|4PSK.1000.167|8PSK.1000.167|4FSK.2000.600S|4FSK.2000.600|4FSK.2000.100|4PSK.2000.100|8PSK.2000.100|4PSK.2000.167|8PSK.2000.167|> *(note high baud rate FM modes in development are not currently included in this list)*

Sets the modulation mode and bandwidth for FEC (broadcast/multicast) transmission. Details on the specific frame types can be found in the ARDOP frame type spreadsheet. In general the first component of the frame ID is the modulation type e.g. 4FSK, 8PSK etc. The second is the bandwidth in Hz (@-26 dB). The third is the baud rate. Some modes also allow a shortened frame designated by a trailing “S”. The above are in order of robustness for most channel types. Note baud rates over 300 are not currently permitted in the US below 29 MHz.

**FECREPEATS** <0-5> Sets the number of times a frame is repeated in FEC (multicast) mode. Higher number of repeats increases good copy probability under marginal conditions but reduces net throughput.

**FECSEND** <True, False>

Start/Stop FEC broadcast/multicast mode for specific FECMODE. FECSEND <False> will abort a FEC broadcast.

**GRIDSQUARE** <4, 6 or 8 character grid square>

Sets or retrieves the 4, 6, or 8 character Maidenhead grid square (used in ID Frames) an improper grid square syntax will return a FAULT.

**INITIALIZE** Clears any pending queued values in the TNC interface. Should be sent upon initial connection and before any other parameters are sent.

**LEADER**<100-2000>

Get/Set the leader length in ms. (Default is 160 ms). Rounded to the nearest 20 ms.

**LISTEN** <False|True>

Enables/disables server’s response to an ARQ connect request to MYCALL or any of MYAUX call signs. Default = True. May be used to block connect requests during scanning or periods when server is offline.

**MONITOR** <False|True>

Enables/disables monitoring of FEC or ARQ Data Frames, ID frames, or Connect request in disconnected ARQ state. Default=True

**MYAUX <aux call sign1, aux call sign2, ... aux call sign10>**

Sets up to 10 auxiliary call signs that will answer ARQ connect requests. Call signs must be valid radio call signs and separated by commas. If sent with an illegal call sign (e.g. "MYAUX x" it will clear the MYAUX list. If sent without a parameter will return a comma delimited string of current MYAUX call signs. Legitimate call signs include from 3 to 7 ASCII characters (A-Z, 0-9) followed by an optional "-" and an SSID of -0 to -15 or -A to -Z. An SSID of -0 is treated as no SSID.

**MYCALL <call sign>**

Sets current call sign. If not a valid call generates a FAULT. Legitimate call signs include from 3 to 7 ASCII characters (A-Z, 0-9) followed by an optional "-" and an SSID of -0 to -15 or -A to -Z. An SSID of -0 is treated as no SSID.

**PLAYBACK <device name>**

Sets desired sound card playback device. If no device name will reply with the current assigned playback device.

**PLAYBACKDEVICES**

Returns a comma delimited list of all currently installed playback devices.

**PROTOCOLMODE<ARQ|FEC>** Sets/Gets the protocol mode. If ARQ and LISTEN above is TRUE will answer Connect requests to MYCALL or any call signs in MYAUX. If FEC will decode but not respond to any connect request.

**PURGEBUFFER** Clears any data in the outbound buffer. Should precipitate a "BUFFER 0" asynchronous response. Data purged goes into a TNC buffer than can be restored to the outbound buffer with a RESTOREBUFFER command.

*The following RADIO commands support optional radio control that may be used in some applications (e.g. Virtual TNC is running on a remote computer located with the radio). All radio commands begin with "RADIO".*

**RADIOANT<0|1|2>** Selects the radio antenna 1 or 2 for those radios that support antenna switching. If the parameter is 0 will not change the antenna setting even if the radio supports it. If sent without a parameter will return 0, 1 or 2. If RADIOCONTROL Is false or RADIOMODEL has not been set will return FAULT

**RADIOCTRL<True|False>** Enables/disables the radio control capability of the ARDOP\_Win TNC. If sent without a parameter will return the current value of RADIOCONTROL enable.

**RADIOCTRLBAUD<1200-115200>** . (Note: baud rates >4800 recommended for PTT control due to T>R Latency)

**RADIOCTRLDTR<True|False>** Enable/disable DTR Line on Control port.

**RADIOCTRLPORT<COMn>** Set/get the radio control com port to use for radio control.

**RADIOCTRLRTS<True|False>** Enable/disable RTS Line on Control port.

**RADIOFILTER<0|1|2|3>** This sets (for selected radios) the pre setup filter for the radio. A value of 0 will disable filter control. Values 1, 2, or 3 will setup the first second or third filters in the radio. Some radios will not support 3 filters. Filters should be setup on the radio to be centered on 1500 Hz and the appropriate bandwidth for the requested Bandwidth. 0 disables filter control.

**RADIOFREQ<Frequency in Hz>** If Radio Control is enabled in the ARDOP\_Win TNC sets the *Dial* frequency of the radio and the display of the Waterfall or Spectrum display. If sent without parameters will return the current *Dial* frequency of the radio.

**RADIOICOMADD<00-FF>** Sets/reads the current Icom Address for radio control (Icom radios only). Values must be hex 00 through FF

**RADIOISC<True|False>** Enable/Disable Radio's internal sound card (some radios)

**RADIOMENU<True|False>** Enable/Disable the Radio menu item on the TNC Form.

**RADIOMODE<USB,USBD, FM>** Sets the radio modulation mode to USB, USB Digital (some radios) or FM (some radios). If sent without a parameter will return the current value of RADIOMODE.

**RADIOMODEL<Radio Model>** If radio control is enabled accepts the radio model. If sent without a parameter returns the current radio model. If radio not supported returns Fault.

**RADIOMODELS<>** Returns a comma delimited list of supported radio models.

**RADIOPTT<CATPTT|VOX/SIGNALINK|COMn>** Selects CATPTT,VOX(SignalLink) mode or COM port for PTT Control

**RADIOPTTDTR<True|False>** Enables/disables PTT keying using DTR signal on RADIOPTT Com port

**RADIOPTTRTS<True|False>** Enables/disables PTT keying using RTS signal on RADIOPTT Com port

**RADIOTUNER<True|False>** Enable/Disable internal radio tuner (some radios). (not currently implemented)

*End of Radio Commands.*

**RESTOREBUFFER**<> Restores data from the most recent PURGEBUFFER command. Should precipitate an asynchronous “BUFFER nnnn” response.

### **SENDID**

This will send an ID frame and if CWID above is enabled followed by a FSK CW ID. The protocol must be in the DISC state or a <Fault: Not from state...> will be replied.

**SETUPMENU**<**True**|**False**> Enabled/Disable the Setup Menu on the ARDOP\_Win TNC main form.

### **SQUELCH** <1-10>

Returns or sets the current squelch value (default = 5). The default value should be sufficient for most installations. Squelch affects the sensitivity of the leader and busy detectors (low values = higher sensitivity but increased false triggering).

### **STATE**

Gets the current ARDOP protocol state  
<**OFFLINE**|**DISC**|**ISS**|**IRS**|**IRS**to**ISS**|**QUIET**|**FEC**Send|**FEC**Rcv)> Every State change is also reported asynchronously with the **NEWSTATE** reply below.

**TRAILER**<**0**-**200**> Get/Set the trailer length in ms. (Default is 0 ms). Rounded to the nearest 20 ms. Normally not required except for some SDR Radios.

**TUNERANGE**<**0**|**50**|**100**|**150**|**200**> Get/set the DSP tuning range in Hz. 0 should only be used in FM or AM radio connections.

### **TWOTONETEST**

Send 5 second two-tone burst at the normal leader amplitude. May be used in adjusting drive level to the radio. If sent while in any state except DISC will result in a fault “not from state .....”

### **VERSION**

Returns the name and version of the ARDOP TNC program or hardware implementation.

## **6.0 Asynchronous Responses**

The ARDOP TNC codec will respond on the command port with possible asynchronous responses. All asynchronous responses begin with “**c: and terminate in <Cr> + 2 Byte CRC**” The host must acknowledge an asynchronous response with a **RDY** or a **CRCEFAULT** response

*This is the list of the current responses:*

**BUFFER**

Reply format is:

BUFFER <data out queued>

Value is in integer bytes. BUFFER may also be polled using the BUFFER command with no parameters.

**BUSY FALSE**

Clear channel detected

**BUSY TRUE**

Busy channel detected

**CANCELPENDING** Indicates to the host that the prior PENDING Connect Request was not to MYCALL or one of the MYAUX call signs) This allows the Host to resume scanning.

**CONNECTED <remote Call Bandwidth in Hz>**

e.g. "CONNECTED W1ABC 500" An ARQ connection has been established. <remote Call> contains the connected call sign with a negotiated bandwidth. This follows the TARGET <target call sign> asynchronous response if the requested bandwidths are compatible and the connection negation is complete.

**DISCONNECTED**

An existing ARQ link has been disconnected

**FAULT <description>**

A program fault or error condition.

**FREQUENCY<Frequency in Hz>** If TNC Radio control is enabled the FREQUENCY command is sent to the Host upon a change in frequency of the radio. This can be caused by either a new programmed frequency, a Dial turn on radios with frequency read back or a mouse click on the Waterfall or Spectrum displays (in USB or USB D modes causing a retune). Does *not* return the new frequency when the command RADIOFREQ sent to the modem. The frequency reported is the *DIAL* frequency of the radio.

**NEWSTATE** reports any protocol state change.

Reply options: < OFFLINE|DISC|ISS|QUIET|IRS|IRS to ISS|FECSend|FECCrv>

**PENDING**

Indicates to the host application a Connect Request frame type has been detected (may not necessarily be to MYCALL or one of the MYAUX call signs). This provides an early warning to the host that a connection may be in process so it can hold any scanning activity.

**PTT <True|False>**

Indicates to the host application to key the PTT on (PTT True) or off (PTT False)

To operate correctly the transmitter PTT should be activated within 50 ms of receipt of this response. Excessive delay in PTT application or removal may cause a failure in ARQ modes if measured latency exceeds 250 ms. Not sent if local TNC radio control is used.

**REJECTEDBW** <**Remote Call sign**> Used to signal the host that a connect request to or from Remote Call sign was rejected due to bandwidth incompatibility

**REJECTEDBUSY** <**Remote Call sign**> Used to signal the host that a connect request to Remote Call sign was rejected due to remote channel busy detection.

**STATUS** <**status text**> Used to signal information text to user.  
e.g. "STATUS CONNECT WITH W1AW FAILED". The syntax of STATUS commands is not rigidly controlled so if used should be for user display purposes only.

**TARGET** <**target call**>

Identifies the target call sign of the connect request. The target call will be either MYC or one of the MYAUX call signs. This command precedes the CONNECTED <remote Call> asynchronous response.

**TUNE** <**Tuning offset in integer Hz**> Sent when the waterfall or spectrum is clicked and can be used by the host to adjust radio settings. This is bounded to approx. +/- 1200 Hz. If radio control is enabled will also tune the radio to the new frequency.

## 7.0 Data transfer:

Data is transferred on the same TCP address/port, Serial COM port or Bluetooth pairing as commands using binary byte transfers with a specific header defined below.

The format for data going from the Host to the TNC is:

**"D:" +2 byte count (Hex 0001 - FFFF most significant byte first), binary data,+2 Byte CRC"**. For TCPIP interfaces the 2 Byte CRC is not used since CRC and repeats are integral to the TCPIP protocol. The **"D:"** identifies this as a data frame from the Host. If the TNC received the data correctly it replies with **"c:RDY<Cr> + 2 byte CRC"** or if a CRC failure **"c:CRCFail<Cr> + 2 byte CRC"**. (No CRC on TCPIP interface commands) The TNC will buffer this data in its outbound queue until transmitted. A failed CRC should prompt the Host to resend the frame up to 3 times before declaring an interface fault.

The format for data going from the TNC to the Host is:

**"d:" + 2 byte count (Hex 0001 - FFFF most significant byte first), + "ARQ|FEC|ERR|IDF", + binary data, +2 Byte CRC"**. For TCPIP interfaces the 2 Byte CRC is not used since CRC and repeats are integral to the TCPIP protocol. The **"d:"** identifies this as a data frame from the TNC. **"ARQ"** indicates this is data received from a connected (ARQ) session and should be error free. **"FEC"** indicates this is data received from an unconnected session ( a monitored FEC or monitored ARQ frame) and should be error free. **ERR** indicates this is Data received

from an unconnected session which failed CRC check of the received frame . It can be masked or ignored by the Host if desired. "IDF" indicates this is ID frame information If the Host received the data correctly it replies with "C:RDY<Cr> + 2 byte CRC" or if a CRC failure "C:CRCFAIL<Cr> + 2 byte CRC). The TNC will then respond with the next data frame or a repeat of the failed data frame. For TCPIP interfaces the 2 Byte CRC is not used since CRC and repeats are integral to the TCPIP protocol.

**Appendix B: ARDOP Channel Busy and Busy Lock Details (place holder...)**