**MACROS**

# Randy Tipton, WA5UFH, Geoff Anderson, G3NPA, Jim Hargrave, W5IFP and Hew Lines VA7HU

## 1.0 GENERAL

Logger32 is user-friendly and can be customized to your own personalized likes by using the program's built-in Macro commands. These commands can be used to program up to 36 programmable buttons in the [Sound Card Data window](#_topic_SoundCardDataWindow1), the [Data Terminal window](#_topic_DataTerminalWindow) and the [CW Machine](#_topic_CWMachineWindow).

This topic is intended to be a reference document for details on how to use the Macro language. Specific details on programming buttons, selecting the number of buttons to display and some suggested functionality to assign to those buttons for each of the windows can be found in the topics [Hot Keys](#_topic_HotKeys) and [Programmable Buttons](#_topic_ProgrammableButtons).

## 2.0 MACRO COMMANDS

A number of these commands require a radio controlled by Logger32 via a serial port for their effective operation. These commands are colored red in the table below. For those who operate a software controlled radio, please see the section "Macros for Software Controlled Radios".

**Note**: The following list details the Macro commands for the individual Logger32 modules and modes.

* + **CW** applies to the [CW Machine window](#_topic_CWMachineWindow)
  + **DATA** applies to the [Data Terminal window](#_topic_DataTerminalWindow)
  + **RCP** applies to the [Radio Control Panel](#_topic_RadioControlPanelRCP)
  + **PSK** applies to the [Sound Card Data window](#_topic_CommunicatingwithPSKMFSKandQPSK)
  + **MMTTY** applies to the [Sound Card Data window](#_topic_CommunicatingwithRTTY)  and all profiles in the MMTTY Engine
  + **MMVARI** applies to the [Sound Card Data window](#_topic_CommunicatingwithRTTY)  and all modes in the MMVARI Engine
  + **LH (Local Host)** applies to the [Telnet Cluster window](#_topic_TelnetClusterWindow)

**$afc$ CW MMTTY MMVari**

Toggles the [AFC](#AFC) function on and off.

**$align$                                                MMTTY MMVari**

Adjusts the radio frequency so the captured signal appears at the default audio frequency. Details of the operation of the align function using mouse clicks are contained in the [Sound Card Data window](#9.2_Align) section.

**$ascii(n)nnn$ Data MMTTY MMVari**

Inserts the [ASCII](#ASCII) code ((n)nnn) of a character than cannot be entered from a keyboard into the output text string (if the first digit is a zero, it is optional.) See the section on [More Difficult Macros.](#7.0_MORE_DIFFICULT_MACROS)

**$band$ CW Data MMTTY MMVari Local Host**

Inserts the current band displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string.

**$bookmark$ CW Data MMTTY MMVari**

Inserts a [Pseudo DXSpot](#_topic_PseudoDXSpots) into the [DX Spots window](#_topic_DXSpotsWindow1) and the [Bandmap](#_topic_TheBandMap).

**$call$ CW Data MMTTY MMVari Local Host**

Inserts the callsign currently displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string.

**$callchanged$               CW**

This Macro is the same as [$call$](#$call$), but only sends something if the call has changed since it was last sent. This Macro is useful when correcting an erroneous callsign

$callchanged$ will not work standalone, it must be embedded with other data. If the Macro is not populated, then the Macro and the space trailing the Macro is erased from the string to be sent.

**Example**: Create a Macro "$callchanged$" 1234. In the [CW Machine](#_topic_CWMachineWindow), enter a callsign of AA1A and transmit the Macro. It should send "AA1A 1234". Transmit the Macro again. It should send 1234. Change the callsign to "AA1AA". Transmit the Macro should send "AA1AA 1234".

**$callsignbeforeclearlog$ CW         MMTTY               MMVari**

This Macro can be used in conjunction with [$clearlog$](#$clearlog$) to respond to dupes during contests or DX expeditions. It will insert the Callsign in the Logbook Entry window into the Macro and clear the [Logbook Entry window](#_topic_LogbookEntryWindow).

Typical use: [$transmit$](#$tune$) [$clearlog$](#$clearlog$) [$callsignBeforeClearLog$](#$callsignbeforeclearlog$) QSO B4 TU QRZ [$receive$](#$receive$)

**$CallsignGetsFocus$ CW**

At the end of a transmission, sets the focus on the Callsign field in the [CW Machine window](#_topic_CWMachineWindow).

**$cancelscrolllock$                 MMTTY                        MMVari**

Has been removed.

**$clear$ CW Data MMTTY MMVari**

An immediate command to clear the TX Buffer and callsign panel. This Macro only works if the keyer is in a Manual TX. [see [Note 2](#Note_2)].

**$clearbuffers$ MMTTY**

Clears the TX Window and Buffer as well as the RX Window. Stops transmission but DOES NOT return to RX. [see [Note 2](#Note_2)]

**$clearlog$ CW Data RCP MMTTY MMVari**

Clears the [Logbook Entry window](#_topic_LogbookEntryWindow).

**$cleartxbuffer$ MMTTY**

Clears the TX Buffer. [see [Note 2](#Note_2)]

**$ClearCallsignOnQSYOn$ RCP**

Checks the Clear Callsign On QSY menu item.

**$ClearCallsignOnQSYOff$ RCP**

Unchecks the Clear Callsign On QSY menu item.

**$ClearQSYMarker$ RCP**

Clears the current Clear Callsign On QSY frequency from memory.

**Control\_<x> Data**

Where <**x**> is a control character (usually an upper case letter to be sent to the TNC to issue whatever command is needed). [**Note**: there are no "$" symbols enclosing this macro].

**$command$ CW Data MMTTY     MMVari**

Sends the text string following the command to a radio controlled by Logger32. **Note**: "xxxx" is a string.

If used from the [CW machine window](#_topic_CWMachineWindow), it is recommended that only one control command be given at a time. To allow multiple commands from a single Macro, the CW Machine will accept a format of $command 1234 / 5678 / 91011$. The "/" is used as a command separator. (See [Note 8](#Note_8))

If used under [SO2R](#_topic_SingleOperatorTwoRadiosSO2RSuppo) conditions, the Macro can accept the format $command[radio1text]|[radio2text]$ where [text] is an optional parameter and the "[" and "]" are NOT part of the parameter. Further information on this can be found in the [Single Operator Two Radios](#_topic_SingleOperatorTwoRadiosSO2RSuppo) topic.

As the TEN-TEC OMNI VII command set requires both ASCII characters and hex numbers, a further modifier has been introduced. Hex numbers can be passed in this Macro if enclosed in "<" and ">". Thus a Macro such as $command \*A<xx>$ where xx is a hex number is acceptable. For more information see the [TEN-TEC OMNI VII](#_topic_TENTECOMNIVII) topic.

**$cw$ MMVari**

Sends the text string following the command  (up to 40 characters) on CW, then switches to RX.

**$cwid$ MMVari**

Transmits a CW ID string at the end of the current transmission, then switches to RX.

**$CWspeeddn$ CW**

Permanently decreases the keying speed by one [WPM](#WPM).(See [Note 2](#Note_2)).

**$CWspeedup$ CW**

Permanently increases the keying speed by one [WPM](#WPM).(See [Note 2](#Note_2)).

**$CWTextGetsFocus$ CW**

At the end of a transmission, sets the focus on the Text field in the [CW Machine window](#_topic_CWMachineWindow).

**$delay$ Data**

Pauses 250 [ms](#ms) before sending the next character to the [TNC](#TNC). This is typically used after sending control codes to the [TNC](#TNC) to change modes.

**$down$ MMVari**

Moves the main passband down 1 Hz. If [AFC](#AFC) is locked on, this command will have no effect.

**$endtime$ Data MMTTY MMVari**

Records the current time as the QSO end time.

**$escape$ Data**

Sends the ESC character (chr$(27)) to the [TNC](#TNC).

**$file$ Data MMTTY MMVari**

When followed on the same line with a fully qualified textfile name ($file$C:\LOGGER32\MYFILE.TXT), this command will load the contents of that file into the TX buffer. If only the filename is used, the command will default to the Logger32 directory. **Note**: The [$file$](#$file$) Macro in the [Data Terminal window](#_topic_DataTerminalWindow) does NOT show the text being loaded into the RX field in the window until load is complete. This is necessary because the software must make sure that any <**Esc**> characters that may be embedded in the text are stripped out and that an appropriate delay is inserted into the string. The same applies to any [Control\_<x>](#Control_<x>) strings in the text.

**$greeting$ CW Data MMTTY MMVari**

Send a greeting appropriate to the local time of the QSO partner. If Logger32 cannot determine the distant end time, then the default greeting will be sent. See The [Greeting Macro](#4.0_THE_GREETING__MACRO) section of the [Programmable Buttons](#_topic_ProgrammableButtons) topic.

**$hexbytes$ Data**

Sends the hexadecimal string following the command to the Data Terminal, Antenna switch port.

**$hexcommand$ CW Data MMTTY MMVari**

Sends the hexadecimal string following the command to the radio. When used from the [CW Machine window](#_topic_CWMachineWindow), it is recommended that only ONE command be issued. To allow multiple commands from a single the CW Machine will accept a format of $hexcommand 00 01 02 03 / 04 05 06 07 / 08 09 10$ The / is used as a command separator. (See [Note 8](#Note_8)).

**$IcomVFOB$ RCP**

Reads and displays VFO B on late model ICOM radios when in split operation..

**$lasthighlightedcall$ MMTTY MMVARI MMTTY**

Places the last highlighted call into the [Logbook Entry window](#_topic_LogbookEntryWindow).

**$lasthighlightedcallandlookup$ MMTTY MMVARI MMTTY**

This Macro is the same as [$lasthighlightedcall$](#$lasthighlightedcall$), and in addition does a callsign lookup (QRZ, Hamcall or whatever the user has configured).

**$lastloggedcallsign$ MMTTY MMVARI**

Use it like $call$, but it inserts the callsign of the last QSO in the logbook into the macro.

You must have logged a callsign (since starting Logger32) for it to work.  It doesn't actually look in the logbook for the callsign, it just remembers the callsign you logged.

**$lastloggedname$ MMTTY MMVARI**

Inserts the Name of the last logged QSO. You must have logged a callsign (since starting Logger32) for it to work.   It doesn't actually look in the logbook for the name, it just remembers the last name you logged.

**$lastqsoband$ Data MMTTY MMVari**

Inserts the last QSO band for the station being worked into the output text string.

**$lastqsodate$ Data MMTTY MMVari**

Inserts the last QSO date for the station being worked into the output text string in the same format as that of the log.

**$lastqsotime$ Data MMTTY MMVari**

Inserts the last QSO time for the station being worked into the output text string in the same format as that of the log.

**$lastqsomode$ Data MMTTY MMVari**

Inserts the last QSO mode for the station being worked into the output text string.

**$log$ CW Data MMTTY MMVari**

Enters the data currently displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the Log. **Note**: $log$ is not a standalone Macro but is to be used embedded into text Macros.

**$logimmediate$ CW MMTTY MMVari**

Logs the QSO and clears the buffer immediately.

**$longdate$ Data MMTTY MMVari**

Inserts the current UTC date into the output text string in the form dd-mm-yyyy.

**$longdateandtime$ Data MMTTY MMVari**

Inserts the current UTC date and time into the output text string in the form dd-mm-yyyy, hh:mm:ss.

**$longtime$ Data MMTTY MMVari**

Inserts the current UTC time into the output text string in the form hh:mm:ss.

**$lookup$ CW**

This Macro executes immediately regardless of where in the Macro it is placed, and triggers the Logger32 auto-lookup (if configured and there is a call in the Callsign edit box of the [Logbook Entry window](#_topic_LogbookEntryWindow)}.

**$loop$ CW MMTTY MMVari**

Causes a five (5) second delay after the Macro has completed, then restarts the Macro (see [More Difficult Macros](#7.0_MORE_DIFFICULT_MACROS)).

**$loop x$ CW MMTTY MMVari**

Causes a user defined delay after the Macro has completed, then restarts the Macro. The value "x" is in seconds.

**$mouseTF-Set$ CW RCP MMTTY MMVari**

For Kenwood radio sends TS1; or TS0; command corresponding to TF-Set state when button is pressed/ held and released by mouse or function key. It needs to check correspomding option in Setup Radio table.

**$mode$ Data MMTTY MMVari**

Inserts the mode currently displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string.

**$msg(x)$ CW (WinKey2 ONLY)**

Macros $msg1$ to $msg6$ are for sending WinKey2 internal memories. These Macros are valid for a WinKey2 only and are immediate Macros.

**$MultiRX$ MMVari**

This Macro activates the multiple receive channels of the MMVari engine. See the [MultiRX](#_topic_MultiRX) topic.

**$mycall$ CW Data MMTTY MMVari**

Inserts the current "operator" value into the output text stream. Usually this would be a callsign.

**$name$ CW Data MMTTY MMVari**

Inserts the contents of the Name field currently displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text stream. For specific use in the [CW Machine](#_topic_CWMachineWindow), see [Note 5](#Note_5) in the CW Machine topic.

**$NameGetsFocus$ CW**

At the end of a transmission, sets the focus on the Name field in the [CW Machine window](#_topic_CWMachineWindow).

**$net$ MMTTY MMVari**

Toggles the [Net](#5.7_Net_On/Off) function on or off

**$numqsos$ Data MMTTY MMVari**

Inserts the total number of QSOs logged for the station being worked.

**$qsx+/-nnnn$ MMTTY MMVari**

Shifts the transmit frequency of the radio up or down by nnnn Hz. $qsx-2000$ shifts the frequency transmit frequency down by 2 kHz, $qsx+1500$ shifts it up by 1.5 kHz. This shift is terminated by the [$simplex$](#$simplex$) command.

**$qsy(nn)nnn.(nn)$ Data MMTTY MMVari**

Immediately changes the radio frequency to the specified value (nnnnn.nn). For example - $qsy7070$ or $qsy14070.22$. See [Note 1](#Note_1).

**$qth$ MMTTY MMVari**

Inserts the contents of the QTH field currently displayed in the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string.

**$radioandtone$ MMTTY MMVari**

Returns the RF center frequency (radio carrier frequency plus audio tone frequency) when in [PSK](#PSK31/63/125) mode. Returns the RF frequency of the Mark tone (radio frequency plus the mark tone audio frequency) in RTTY mode (see the section on [More Difficult Macros](#7.0_MORE_DIFFICULT_MACROS)). See also [Note 5](#Note_5).

**$radio1$ RCP**

Selects Radio #1. Does nothing if Radio #1 is already active.

**$radio2$ RCP**

Selects Radio #2. Does nothing if Radio #2 is already active

**$radio1->radio2$ RCP**

Transfers frequency and mode of Radio 1 to radio 2

**$radio2->radio1$ RCP**

Transfers frequency and mode of radio 2 to radio 1,

**$Radio1Offset xxxxx$ RCP**

xxxxx represents the offset/transverter offset in kHz. For example, to concert radio 1 from 28MHz to 144Mhz, the macro would be $Radio1Offset 116000$. To turn off the transverter offset the macro would be $Radio1Offset 0$.

**$Radio2Offset xxxxx$ RCP**

xxxxx represents the offset/transverter offset in kHz. For example, to concert radio 2 from 28MHz to 144Mhz, the macro would be $Radio2Offset 116000$. To turn off the transverter offset the macro would be $Radio2Offset 0$.

**$radiocontrolpanelmacroxx$  CW MMTTY MMVARI**

Allows execution of RCP macros while retaining Sound card or CWM focus. Where xx is 1 to 36. (For the 36 macro buttons)? Typical use would be to execute a RCP macro that increments the radio by 1 kHz for RTTY and PSK Sprint operation.

**Note**: This macro executes RCP macros assigned to the radio of focus. If you have 2 radios be sure the desired function is in the same macro slot for both radios. Otherwise you need to have a separate macro in the CWM/Sound card for each radio.

**Note**: The RCP panel does not have to be open for this macro to function.

**$radiofreq$ Data MMTTY MMVari**

For radios controlled by Logger32, returns the nominal RF carrier frequency of the radio (see the section on [More Difficult Macros](#7.0_MORE_DIFFICULT_MACROS)).

**$receive$ CW MMTTY MMVari**

Switches the radio from transmit to receive.

**$receivedrst$ CW Data MMTTY MMVari**

Inserts the contents of the RST received field of the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string. The default is 599 if nothing is entered.

**$receivedrstn$ CW**

Inserts the contents of the RST received field of the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text string. Any figure 9 is sent as the character "N". The default is 5NN if nothing is entered.

**$receivedgrid$ Data MMTTY MMVari**

Inserts the received Grid Square, if entered, into the [Logbook Entry window](#_topic_LogbookEntryWindow).

**$reset$ CW**

An immediate command to reset the parallel port or the serial port CW keying and/or the PTT control lines to a low state (unkey the radio). [See [Note 2](#Note_2) below]

**$rotor$ CW Data MMTTY MMVari**

This Macro rotates the antenna to the computed short path azimuth after a callsign is entered into the [Logbook Entry window](#_topic_LogbookEntryWindow). It is the same as the <**CtrlL+A**> key combination in that window.

**$rotorlp$ CW Data MMTTY MMVari**

This Macro rotates the antenna to the computed long path azimuth after a callsign is entered into the [Logbook Entry window](#_topic_LogbookEntryWindow). It is the same as the <**Alt+A**> key combination in that window.

**$rtty$ MMTTY MMVari**

Changes the operational mode of the [Sound Card Data window](#_topic_SoundCardDataWindow1) to MMTTY "Normal Mode".

**$rtty-i$ MMTTY MMVari**

Changes the operational mode of the [Sound Card Data window](mk:@MSITStore:C:/Ham%20Radio/Logger32/Help/Draft/Ver%2090.chm::/to%20do/sound%20card%20data%20window.htm) to [MMTTY](mk:@MSITStore:C:/Ham%20Radio/Logger32/Help/Draft/Ver%2090.chm::/to%20do/sound%20card%20data%20window.htm) “Reverse” mode  
  
**Note**: If a special MMTTY profile has been selected, this macro will switch to MMTTY and open the profile last used, retaining the selected mode. If the last used was the one of the default profiles, it will open to that profile in reverse mode..

**$rttybpfoff$ MMTTY MMTTY only**

Turns the receive bandpass filter off.

**$rttybpfon$ MMTTY MMTTY only**

Turns the receive bandpass filter on

**$rttyfigures$ MMTTY MMTTY only**

Shifts the RTTY transmission into Figures mode.

**$rttyletters$ MMTTY MMTTY only**

Shifts the RTTY transmission into Letters mode.

**$rttymarkfrequency$ MMTTY MMTTY only**

Returns the RTTY Mark frequency. This frequency will change depending on the commands [$rttynormal$](#$rttynormal$) or [$rttyreverse$](#$rttyreverse$).

**$rttynormal$ MMTTY MMTTY only**

Operate RTTY using normal shift (LSB).

**$rttyreverse$ MMTTY MMTTY only**

Operate RTTY using reverse shift (USB).

**$rttysetup$ MMTTY MMTTY only**

Display the MMTTY Setup dialog box. This is the equivalent of selecting the "Setup" button of the RTTY Toolbar.

**$rttyshift$ MMTTY MMTTY only**

Returns the current setting of the RTTY Shift (normal or reverse).

**$rttysquelchoff$ MMTTY MMTTY only**

Turns the RTTY Squelch off.

**$rttysquelchon$ MMTTY MMTTY only**

Turns the RTTY Squelch on.

**$rxtonefreq$ MMTTY MMVari**

Returns the Mark tone currently displayed in the second panel of the [Sound Card Data Window Statusbar](#5.0_THE_STATUS_BAR_-_AN_INFORMATION_CONTROL_MECHANISM) (see the section on [More Difficult Macros](#7.0_MORE_DIFFICULT_MACROS)).  See also [Note 5](#Note_5) below

**$selcal$ Data**

Inserts the contents of the SELCAL field (top left corner of the [Data Terminal window](#_topic_DataTerminalWindow)).

**$sentrst$ CW Data MMTTY**

Inserts the contents of the RST Sent field of the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text stream. The default is 599 if nothing is set in the Logbook Entry window.

**$sentrstn$ CW**

Inserts the contents of the RST Sent field of the [Logbook Entry window](#_topic_LogbookEntryWindow) into the output text stream. Any figure 9 is sent as the letter "N". The default is 5NN if nothing is set in the Logbook Entry window.

**$serialnum$ CW Data MMTTY MMVari**

Inserts the serial number of this QSO.

**$serialnum-1$ CW Data MMTTY MMVari**

Inserts the serial number of the previous QSO.

**$shortdate$ Data MMTTY MMVari**

Inserts the current UTC date into the output text string in the form dd/mm/yy.

**$shortdateandtime$ Data MMTTY MMVari**

Inserts the current UTC date and time into the output text string in the form dd/mm/yy hh:mm.

**$shorttime$ Data MMTTY MMVari**

Inserts the current UTC time into the output text string in the form hh:mm.

**$simplex$ MMTTY MMVari**

Restores simplex operation by returning the Transmit frequency to the Receive frequency after a [$qsx$](#$qsx) command.

**$SlavePortOpen$ RCP**

**$SlavePortClose$ RCP**

**$so2r$ RCP**

Opens the [SO2R](#_topic_SingleOperatorTwoRadiosSO2RSuppo) [RCP Panel](#_topic_RadioControlPanelRCP). If it is already open, it does nothing.

**$so2v$ RCP**

Opens the [SO2V](#_topic_SingleOperatorTwoVFOsSO2VSupport) [RCP Panel](#_topic_RadioControlPanelRCP). If it is already open, it does nothing.

**$so2von$ RCP**

Opens the [SO2V](#_topic_SingleOperatorTwoVFOsSO2VSupport) [RCP panel](#_topic_RadioControlPanelRCP).

**$so2voff$ RCP**

Closes the [SO2V](#_topic_SingleOperatorTwoVFOsSO2VSupport) [RCP panel](#_topic_RadioControlPanelRCP).

**Note**: The above 2 Macros can be used standalone or included as part of the split/unsplit Macro commands.

**$speed-$ CW**

Temporarily decreases the keying speed by two WPM. This Macro is temporary, and only has effect within the current transmission. The Macro is reset at the end of the current transmission.

**$speed+$ CW**

Temporarily increases the keying speed by two WPM. This Macro is temporary, and only has effect within the current transmission. The Macro is reset at the end of the current transmission.

**$splitaudioalert$ RCP**

Sounds the split audio alert when the radio is placed into split mode using a manual Macro.

**$splitvisualalerton$ RCP**

Turns on Visual split alert when radio is placed in split mode with a manual Macro command.

**$splitvisualalertoff$ RCP**

Turns off visual alert when radio is placed in simplex using a manual Macro.

**$SRXGetsFocus$ CW**

At the end of a transmission, sets the focus on the SRX field in the [CW Machine window](#_topic_CWMachineWindow).

**$starttime$ Data MMTTY            MMVari**

Records the current time as the QSO start time.

**$srx$ Data MMTTY MMVari**

The content of this field in the [Log Entry window](#_topic_LogbookEntryWindow) will be transmitted.

**$stx$ Data MMTTY MMVari**

The content of this field in the [Log Entry window](#_topic_LogbookEntryWindow) will be transmitted.

**$TF-Set$ CW RCP MMTTY MMVari**

For Kenwood radio sends TS1; or TS0; command corresponding to TF-Set state when button is clicked by mouse or hit by function key. It needs to check correspomding option in Setup Radio table.

**$tncdate$ Data**

Sends the [UTC](#UTC) date and time as yymmddhhmmss.   **Note**: this is actually the computer date/time but it is intended for use in the setting of the TNC time. Some TNCs may require a command to prefix this Macro (e.g. "DATE $tnctime$").

**$toggleradios$ CW Data MMTTY MMVari**

An immediate Macro to toggle the ([SO2R](#_topic_SingleOperatorTwoRadiosSO2RSuppo)) radios in use. The same action as the <**Ctrl+T**> key combination (see [Notes 2](#Note_2) and [4 below](#Note_4)).

**$transmit$ CW MMTTY MMVari**

Turns the transmitter on and transmit any text in the TXWindow

Raises the CW PTT. Can be used when MOX and/or slow typing modes are in use.

**$tune$ CW**

In the [CW machine window](#_topic_CWMachineWindow) this Macro acts as a toggle to transmit a carrier to tune the transceiver. The first instance turns the carrier on, the second instance turns the carrier off.

**$twohexbytes xx xx xx$ Data**

A Macro to send hex characters to the TNC

**$uham xx$ RCP**

Set op command without the need for the <**Cr**> key where xx is the appropriate microHAM control op command.

**$up$ MMTTY MMVari**

Moves the main passband up 1 Hz. If AFC is locked on, this command will have no effect.

**$upperorlower$ MMTTY MMVari**

Returns a plus sign (+) if the radio is in USB mode or a minus sign (-) if the radio is in LSB mode (see the section on [More Difficult Macros](#7.0_MORE_DIFFICULT_MACROS)). See also [Note 5](#Note_5).

**$version$ Data MMTTY MMVari**

Enters the version number of Logger32 into the output text stream.

**$wait x$ RCP**

Inserts a delay time between radio commands. The value "x" is in seconds.

**$winkeymergedletters$ CW**

To send concatenated letters with a Winkey enter a Macro like this ... $winkeymergedletters$BT This sends <**Esc**>BT to the Winkey  For Concatenation of adjacent letters see [Note 3](#Note_3) below.

## 3.0 MODE SELECTION MACROS

**$mmvarimodexx$                                                 MMVari**

Changes the operational mode of the [Sound Card Data window](#2.2_MMVARI_Sound_Card_Selection) to MMVARI modes, where xx is the menu index number of the MMVARI modes as shown in the table below

$mmvarimode0$ GMSK (MBCS)

$mmvarimode1$ FSK (MBCS)

$mmvarimode2$ FSK-W (MBCS)

$mmvarimode3$ BPSK 31 (MBCS)

$mmvarimode4$ BPSK 63 (MBCS)

$mmvarimode5$ BPSK 125 (MBCS)

$mmvarimode6$ BPSK 250 (MBCS)

$mmvarimode7$ BPSK 31

$mmvarimode8$ BPSK 63

$mmvarimode9$ BPSK 125

$mmvarimode10$ BPSK 250

$mmvarimode11$ RTTY-L

$mmvarimode12$ RTTY-U

$mmvarimode13$ MFSK-L 4

$mmvarimode14$ MFSK-L 8

$mmvarimode15$ MFSK-L 11

$mmvarimode16$ MFSK-L 16

$mmvarimode17$ MFSK-L 22

$mmvarimode18$ MFSK-L 32

$mmvarimode19$ MFSK-L 64

$mmvarimode 20$ MFSK-U 4

$mmvarimode21$ MFSK-U 8

$mmvarimode 22$ MFSK-U 11

$mmvarimode 23$ MFSK-U 16

$mmvarimode 24$ MFSK-U 22

$mmvarimode 25$ MFSK-U 32

$mmvarimode 26$ MFSK-U 64

$mmvarimode 27$ QPSK-L 31

$mmvarimode 28$ QPSK-L 63

$mmvarimode 29$ QPSK-L 125

$mmvarimode 30$ QPSK-U 31

$mmvarimode 31$ QPSK-U 63

$mmvarimode 32$ QPSK-U 125

**Notes**:

Modes MFSK-L and MFSK-U have been replaced with MFSK-L xx and MFSK-U xx where xx defines the mode speed. If the original mode name of MFSK-L or MFSK-U was an operational mode for the user and the shortcut keys had been set up then the MMvarisoundcardmacros.ini file will need to be modified to regain the use of the original shortcuts. With Logger32 closed down, open the file MMvarisoundcardmacros.ini and look for [MFSK-L ] or [MFSK-U] including the brackets and change this header to read [MFSK-L 16]] or MFSK-U 16] as appropriate. On reopening Logger32 the original Macro set will appear.

All the mode index numbers greater than 6 have been changed. If the Macro $mmvarimode x$ (where x is an MMVARI mode index number) has been used within a shortcut then it may need to be recoded if it is to point at the correct mode.

## 4.0 ADDITIONAL MACROS

[                                                           **MMTTY**

Turn RTTY diddle off. This command also works if typed into the TXWindow.

]                                                           **MMTTY**

Turn RTTY diddle on. This command also works if typed into the TXWindow.

^          Concatenation of adjacent letters. [see [Note 3](#Note_3)].

## 5.0 MACRO NOTES

**Note 1**: These commands are designed to change the operation of a radio controlled by Logger32. Since these commands change mode, they can be invoked from any other mode. They are ignored if the radio is transmitting so they cannot be used if in a Macro statement that is meant to execute while the radio is transmitting (e.g. "Fred ? QSY to 14080",$RTTY$,$qsy 14080.00$).

**Note 2**: This command can be used ONLY as a standalone Macro. NO other text can be included in the command.

**Note 3**: Concatenated CW letters like AR, SK, BK, etc, are simulated by Macro text with a ^ (caret) between the letters to be joined, i.e., A^R, S^K.

**Note 4**: Macro commands are only available via the buttons. They may not be typed manually in the TX Windows.

**Note 5**: MMVARI shows the RTTY RX freq as the middle of the mark and space (unlike MMTTY).

**Note 6**: For those users who are familiar with Zakanaka, the following Zakanaka Macros have not been included in the Logger32 Sound Card module:

$cd$

$comment$

$page$

$previous$

The following Zakanaka Macros have been re-named in Logger32

$myrst$ to $receivedrst$

$hisrst$ to $sentrst$

This Macro will 'log' the QSO, clear the [Logbook Entry window](#_topic_LogbookEntryWindow) and revert back to receive mode.

73 [$name$](#$name$) TNX for the nice PSK QSO.

Hope to catch you on the 'waterfall' again soon...

[$call$](#$call$) de [$mycall$](#$mycall$) SK SK

[$log$](#$log$)

**Note 7**: This is not a standalone Macro and is to be used embedded in a text Macro. See the example below.

**Note 8**: The [$command$](#$command$) and [$hexcommand$](#$hexcommand$) macros have been expanded for use with SO2R set-ups. Please refer to the [SO2R](#_topic_SingleOperatorTwoRadiosSO2RSuppo) section of the Helpfile.

## 6.0 MMTY MACROS

A few MMTTY macros work in the RTTY mode of the Sound Card Data window:

]                       diddle on

[                       diddle off

~                      pause

%T                    UTC time

%t                     UTC time

%D                   UTC date

## 7.0 MORE DIFFICULT MACROS

This section provides examples of how to use some of the more difficult or unusual Macro commands.

### 7.1 $asciinnn$

This Macro inserts the ASCII code (n)nnn of a character than cannot be entered from a keyboard into the output text string. For example, the ASCII code 191 will display the symbol  "?". So the Macro command $ascii191$Que pasa  will insert "?Que pasa?" into the output text string. If the first digit is a zero, it is optional with this command, you can use either three or four characters.  In other words, $ascii0191$ or $ascii191$, entered in a Macro command, will both generate the character "?".

The character, tilde (~) cannot be typed into a Macro, but if you need the tilde character, use $ascii0126$ in its place, and the tilde character will appear.

**Example 1**:

Set up a Macro button (say F12) that has the command $ascii0176$ and a button caption of chr$(0176) and save it. Simple enough!

Now, with the cursor in the TxWindow, enter the text "The temp here today is 90<F12>F." (Bet you're wondering how I did that ...)

### 7.2 $upperorlower$, $radiofreq$, $rxtonefreq$, and $radioandtone$ Macros

These commands are applicable to both PSK and RTTY.

Here is an example of how to use these them in PSK:

The following text is entered into the text window for Hot Key <**F5**>

Your exact frequency is [$radioandtone$](#$radioandtone$).  Here's how I figure that out ... My receiver is on  [$radiofreq$](#$radiofreq$).  I am receiving your signal at $rxtonefreq$.  Therefore your actual signal is $radiofreq$ $upperorlower$ $rxtonefreq$ = $radioandtone$.

The Button caption is set to "FREQ IN" and the Back color is set to blue.

Pressing the <**F5**> key or clicking on the button titled "FREQ IN" will now insert the following text into the TX Window for transmission:

If you wanted to send the same message using RTTY, you need only change $rxtonefreq$ to $rttymarkfrequency$ in the macro text.

### 7.3 $command$ and $hexcommand$ Macros

These Macros are used to send strings of commands to the radio serial port in either ASCII ($command$) or hexadecimal ($hexcommand$). You can see these codes being sent to the radio in the Radio Debug window.  The topic [Direct Control of Radios](#_topic_DirectControlofRadios) shows how to use these codes for some radios.

Normally, control of the radio is handled by Logger32, but these Macros can be used to implement functions not already present in Logger32. Program them into Macro buttons so you can invoke a command sequence with a script assigned to a single Macro button.  For example, you could program a Macro button to select filters in your radio.

### 7.4 $log$ Macro

This Macro will 'log' the QSO, clear the [Logbook Entry Window](#_topic_LogbookEntryWindow) and revert the system back to receive mode.

73 $name$ TNX for the nice PSK QSO.

Hope to catch you on the 'waterfall' again soon...

$call$ de $mycall$ SK SK

$log$

$receive$

## 8.0 MACROS FOR SOFTWARE CONTROLLED RADIOS

### 8.1 $qsy$ and $command$ Macros

$qsy$ and $command$ Macro support has been added for software controlled radios

The format is $command 14000.123 CW$

The space(s) at the beginning and end of the string are optional. The space(s) between the frequency and the mode are NOT optional. The format of the frequency is in KHz, and the string can use either a "," (comma) or a "." (period) as the decimal separator. The mode must be a valid mode for the radio in use. The only valid $command$ is to change the radio frequency and/or mode.

## 9.0 THINGS TO WATCH OUT FOR

### 9,1 A Warning About Callsigns

When you use Macros to send your callsign at the end of a transmission, especially in RTTY mode, it is wise to put some spaces and the letter K after your callsign. When the receiving station displays or prints your transmission, there may be "garbage" characters that appear immediately after you end your transmission.  This is because the receiving station's squelch could take a second or two to adjust to the no-signal condition.  If your callsign was sent as the last item of text before you turn off the carrier, the other station may add some "garbage" characters to it.  For example, "... de KX2A" may become "... de KX2AP" because a "garbage" P was displayed when the carrier went off.  However, "... de KX2A    K" would be displayed as "...de KX2A    KP," and it is unlikely that the receiving station will confuse the callsign.

### 9.2 System Time Adjustment

If you decide to use a time stamp (some contests even require this) then you may want to be sure that your computer system clock is correct. See the topic [Atomic Clock](#_topic_AtomicClock).