Here in there are two appendixes provided from the MARS documentation that detail how to create a custom radio control driver and MACRO files for MS-DMT.

## APPENDIX M: Man Machine Interface

A Man Machine Interface (MMI) Interpreter engine has been added to MS-DMT similar to that of MARS-ALE.

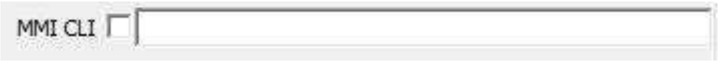
Currently the MMI support provided is for Radio Control only. However in time pretty much all aspects of MS-DMT will be configurable and controllable via MMI scripting.

At this time since we are limited to radio control MMI commands only, all MMI commands are detailed in the next appendix covering DMT-RADI..

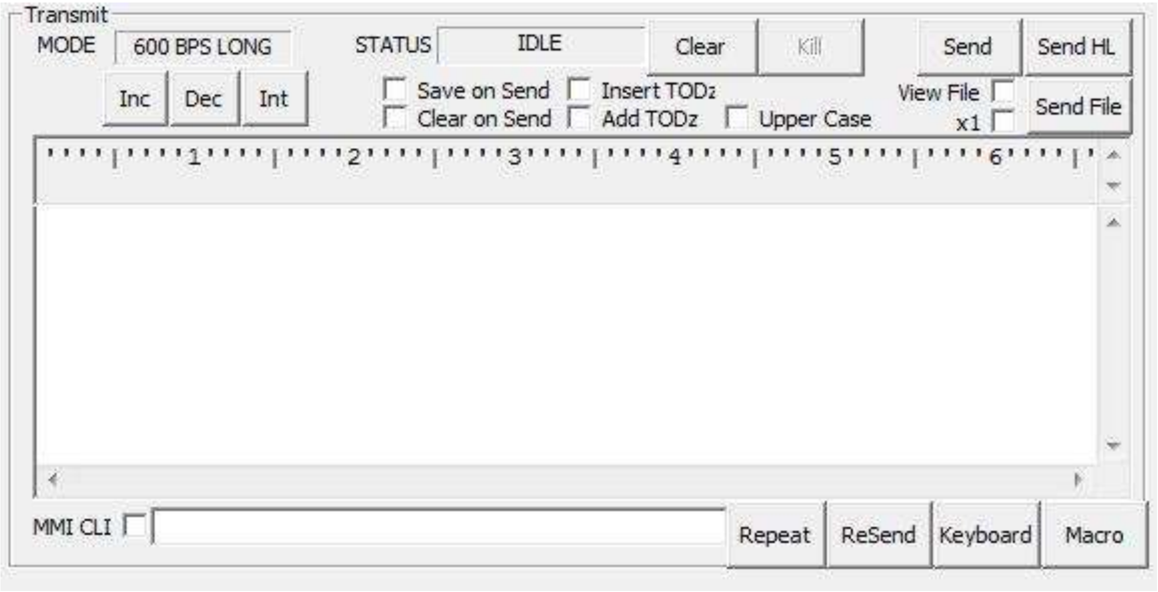
There are five means of MMI command interfaces:

**A.** Command Line Interface (CLI) for single line macro creation when the MMI CLI box is checked to make use of the Peer2Peer message window as the CLI. This is similar to the Data Bar in MARS-ALE.

The state of the MMI CLI checkbox is retained on restart of the tool.



Double Clicking the Peer2Peer message window will clear the window regardless of being in Message mode or MMI CLI mode.


MODE 600 BPS LONG

STATUS IDLE

**B.** CONFIG, STARTUP and SHUTDOWN .DAT files.

CONFIG.DAT is processed first and prior to the radio serial port being opened, thus the radio control MMI commands cannot be used.

STARTUP.DAT is processed next and after the radio serial port is opened.

SHUTDOWN.DATA is processed at normal MS-DMT termination and prior to the radio port being closed.

**Note: As MS-DMT configuration parameters are stored in the DMT.DAT file and as executing MMI commands takes time at program start. I recommend that a .MAC file be use to re-configure MS-DMT when a large number of MMI commands would be used vs. always having them execute at program start.**

**C.** The MACRO (.MAC) files found in the /MACRO/ folder. The MACRO button at MS-DMT GUI bottom. This is the same approach as found in MARS-ALE, where existing radio control .MAC files from MARS-ALE user can be reused "AS IS".

**D.** The DMT-RADIO interface via a radio class of .MAC files found in the /DMT-RADIO/ folder under the location of the MS-DMT executable (.EXE) file.

Full details of DMT-RADIO are provided in the next appendix herein.

**E.** Remote Control: MMI Command Interface -

When RIG = DMT-RADIO or DMT-RADIO_HS the radio can be controlled via an external application via the Remote Control command: <<CMD:MMI CMD>>

Example:

```
<<CMD:MMI CMD>>radcmd TX;
```

```
<<CMD:MMI CMD>>hexradcmd FE FE 8A E0 1A 37 00 01 FD
```

This means that an external application can control the radio to do anything desired.

# APPENDIX N: DMT-RADIO

The DMT-RADIO concept is similar to that of MMI-RADIO in MARS-ALE in that it makes use of a command interpreter as the Man Machine Interface (MMI) that supports user defined macro's using the MMI radio control commands.

The DMT-RADIO support provides the end user a means to create a custom radio control library within MS-DMT. This enables the user to support any make/model of HF SSB radio that supports remote control that has not yet or may never be added to MS-DMT as hard coded radio selections. It also permits the end user to experiment with radio features that are not provided by existing hard coded support for their make/model radio. However it is up to the end user to properly develop and debug support for their specific make/model radio.

The user can develop their own radio control support to a set of interface rules and the remote control syntax for their make/model radio. There is no need to know how to program in any particular computer programming language. Instead the DMT-RADIO concept makes use of the ASCII MMI commands, chief among them being RADCMD and HEXRADCMD and the specific make/model radio's remote control commands to create MACRO's in a library of DMT-RADIO class files. Then at MS-DMT startup, PTT state changes and shutdown, MS-DMT will execute specific MACRO (.MAC) files during its execution in support of required radio control.

The DMT-RADIO specification herein details the specific files that must exist, regardless of their actual need to control a particular HF SSB radio of focus. Some files required are based on the user's configuration selections such as use of CAT for PTT. The required library of radio specific MMI command based MACRO (.MAC) files or collectively the DMT-RADIO Library of files for the particular make/model radio to be controlled must be stored in the \DMT-RADIO\ subdirectory under where the MS-DMT .EXE exists.

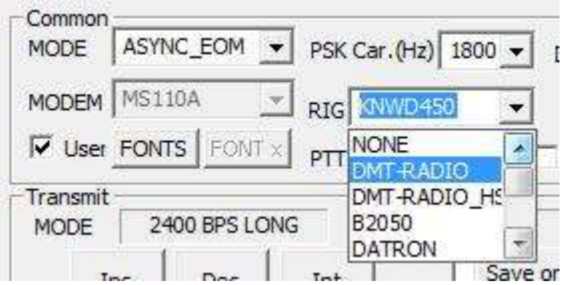
**CAT COM PORT PARAMETERS**

The CAT com port parameters should be selected first.

Regardless of the actual radio type implemented using the DMT-RADIO interface, the default CAT radio port parameters of 9600 N81 will be applied at first. Any other serial radio port parameters can be selected, after which they will be retained in the DMT.DAT file.

**RADIO TYPE DMT-RADIO**

To use DMT-RADIO it or DMT-RADIO_HS must be the radio type selection on the RIG pull down menu as seen below.



Common

MODE ASYNC_EOM

PSK Car.(Hz) 1800

MODEM MS110A

RIG KNWD450

☑ User

Transmit

MODE 2400 BPS LONG

**RADIO ADDRESSING**

Should the actual radio make/mode require Radio Addressing, the MMI Commands used in the various DMT-RADIO_[command class].MAC files shall provide the radio addressing required.

**MMI RADIO COMMANDS SYNTAX**

Command Syntax:

**RADCMD** [Argument1..Argument5]

or

**HEXRADCMD** [Argument1..Argument30]

RADCMD supports radio make/models using an ASCII command set whereas HEXRADCMD supports any command set.

The commands RADCMD and HEXRADCMD can be used in lower or uppercase.

Arguments entered on the same line as RADCMD or HEXRADCMD must delimited by a SPACE character. When the radio protocol supports stacking commands with no spaces then the data string is looked at as one Argument by RADCMD. Due to the nature of HEXRADCMD each byte must be separated by a SPACE character.

Any data that can be sent using RADCMD can be translated to HEX for sending with HEXRADCMD should any special characters not supported by RADCMD be required to control the radio.

To make use of RADCMD or HEXRADCMD the user MUST know the remote control protocol for their particular make/model radio.

RADCMD and HEXRADCMD are used the same as implemented in MARS-ALE, see the section titled “USER MACRO (.MAC) FILES” herein for details.

Command Syntax:

**DELAY nn..nnnn**

Where nn..nnnn supports 10ms to 1000ms (1 second) of timing delay.

Each line with DELAY nn..nnnn command statement will provide for a delay between commands sent to the radio. Thus if the implemented radio make/model requires 100ms pacing between commands the use of DELAY 100 would be made.

Command Syntax:

**CR1**

For use with RADCMD only.

Adds a single Carriage Return to the message sent to the radio when used as an argument to RADCMD.

Command Syntax:

**CR2**

For use with RADCMD only.

Adds a two Carriage Returns to the message sent to the radio when used as an argument to RADCMD.

Command Syntax:

**LF1**

For use with RADCMD only.

Adds a single Line Feed to the message sent to the radio when used as an argument to RADCMD.

Command Syntax:

**LF2**

For use with RADCMD only.

Adds two Line Feeds to the message sent to the radio when used as an argument to RADCMD.

Command Syntax:

**CRLF**

For use with RADCMD only.

Adds a Carriage Return and Line Feed to the message sent to the radio when used as an argument to RADCMD.

**DMT-RADIO MACRO CLASS FILES**

Each of the DMT-RADIO class files will reside in the \DMT-RADIO\ sub directory under the directory containing the MS-DMT .EXE file and will use the master naming convention of:

DMT-RADIO_[command class].MAC

When either DMT-RADIO or DMT-RADIO_HS is selected as the radio type, all of the .MAC class files must exist, even if not used, otherwise a message will be displayed for each missing file.

**DMT-RADIO FILE CLASSES:**

The following files constitute the entire DMT-RADIO class of files.

PTT class files – Always required.

**DMT_RADIO_PTT_ON.MAC**

**DMT_RADIO_PTT_OFF.MAC**

DATA/MIC class files – Radio dependent.

**DMT_RADIO_DATA_ON.MAC**

**DMT_RADIO_DATA_OFF.MAC**

REMOTE state class files – Radio dependent.

**DMT_RADIO_ENABLE_REMOTE.MAC**

**DMT_RADIO_DISABLE_REMOTE.MAC**

RADIO CONFIGURATION class files – User option.

**DMT_RADIO_PRECONFIGURATION.MAC**

**DMT_RADIO_DEINITIALIZE.MAC**

The more DMT-RADIO class files that are used and the more commands that are sent to the radio when using the optional Preconfiguration and Deinitialize class files, the longer it will take for the MS-DMT GUI to appear at program start and remain on screen at normal program shutdown.

**REMOTE ENABLE/DISABLE -**

REMOTE ENABLE/DISABLE MACRO files naming convention:

DMT_RADIO_ENABLE_REMOTE.MAC

DMT_RADIO_DISABLE_REMOTE.MAC

These files support radios that must be commanded to enter and exit radio Remote Control operation such as is required with many older Yaesu, JRC and other older makes of radios.

It should be noted that DMT_RADIO_ENABLE_REMOTE.MAC is executed prior to any radio control being executed. The DMT_RADIO_DISABLE_REMOTE.MAC is executed at normal program shutdown prior to closing the CAT radio port.

If the radio type being used does not require remote enable/disable then just leave them empty.

**PRECONFIUGRATION -**

Preconfiguration MACRO file naming convention:

DMT_RADIO_PRECONFIGURATION.MAC

This file is executed just after opening the radio serial port at the point after DMT-RADIO is initialized.

**DEINITIALIZE -**

Deinitialize MACRO file naming convention:

DMT_RADIO_DEINITIALIZE.MAC

This file is executed at program termination.

**PTT -**

PTT MACRO files naming convention support the Microphone (MIC) port and the DATA (DIG) port that a radio may have where the CAT PTT commands for the ports may differ. At a minimum, if CAT PTT is used, the MIC port PTT files must exist with the correct commands for the radio being implemented.

For the MIC port the PTT class files naming convention is:

MMI_RADIO_MIC_PTT_ON.MAC

MMI_RADIO_MIC_PTT_OFF.MAC

For the DIG port when USB-D or LSB-D are selected as the mode the PTT class files naming convention is:

MMI_RADIO_DIG_PTT_ON.MAC

MMI_RADIO_DIG_PTT_OFF.MAC

These files are only called if CAT PTT is active where the “DIG” are only called if USB-D or LSB-D is selected as the mode for the scan group channel in use.

The RED PTT button on the Data Bar when used, will only use the MIC port for PTT where if USB-D or LSB-D is the mode selected for the given scan group channel, the mode will be switched to USB or LSB respectfully for TX and then set back to USB-D or LSB-D respectfully upon RX.

If there is no specific PTT commands for the given radio being implemented for DIG port use, then the same PTT ON and PTT OFF commands would be entered in both types of PTT class files.

## SYSTEM .DAT FILES

The system .DAT files can be used to process any MMI commands. However the use of RADCMD. HEXRADCMD and other MMI radio control command CANNOT be used in CONFIG.DAT as it executes before the serial ports are opened.

**Note:** At present only the radio control MMI commands are coded for use.

All system .DAT files reside in the same sub directory as the MS-DMT executable (.EXE). They must exist even if not used otherwise a message will be displayed whenever the software is run and their point of execution is reached.

The system .DAT files are:

CONFIG.DAT - Executes at program start prior to any ports or other .DAT files being opened.

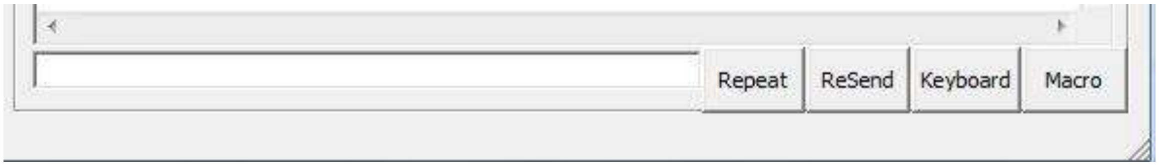
**Note:** As MS-DMT configuration parameters are stored in the DMT.DAT file and as executing MMI commands takes time at program start. I recommend that a .MAC file be use to re-configure MS-DMT when a large number of MMI commands would be used vs. always having them execute at program start.

STARTUP.DAT - Executes MMI commands at program start after radio port is opened.

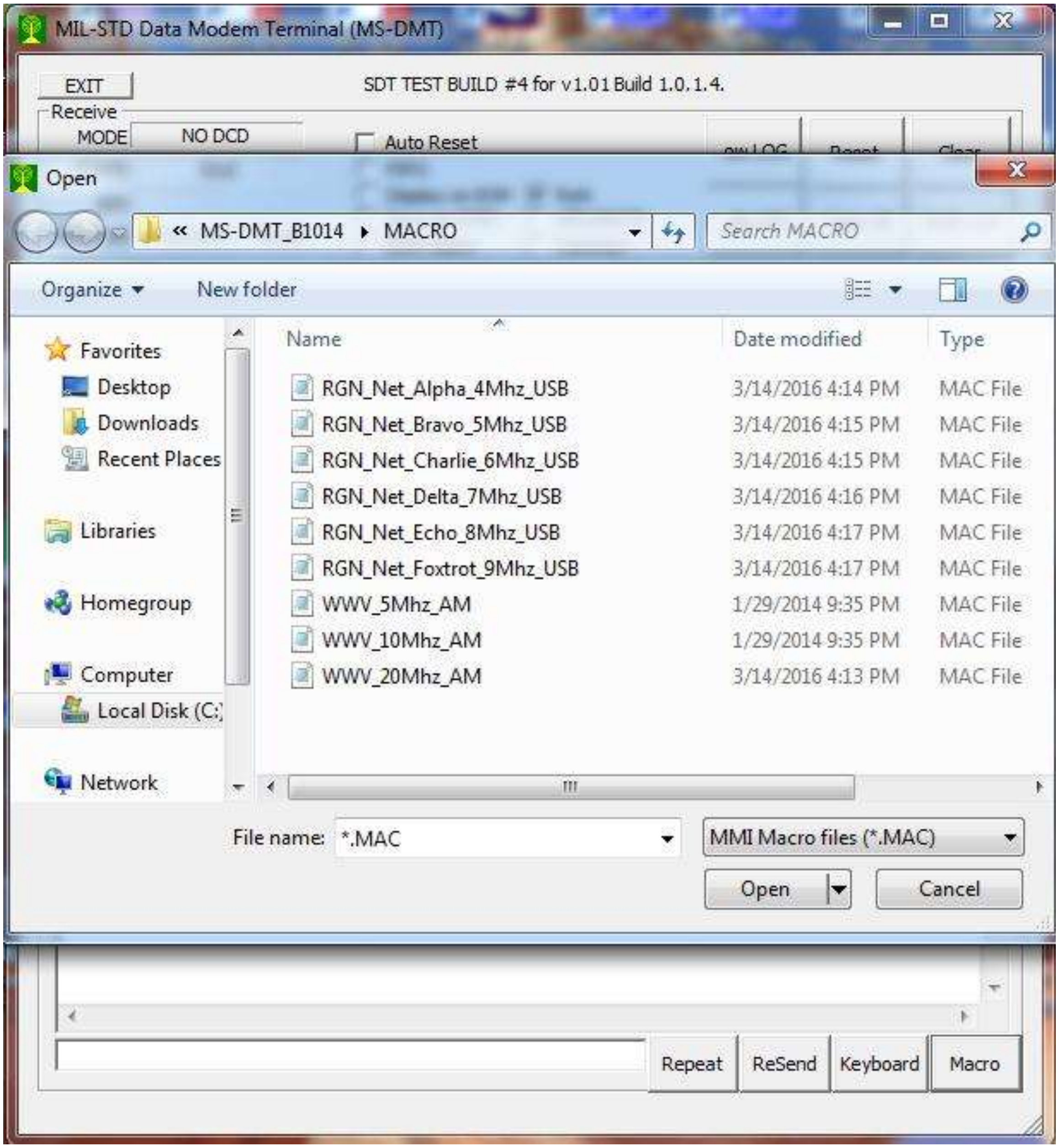
SHUTDOWN.DAT - Executes MMI commands at the termination of the application.

**USER MACRO (.MAC) FILES**

Below are some examples of the use of RADCMD and HEXRADCMD in the creation of MACRO's for CAT radio control purposes by the user via of the MS-DMT MACRO button.



The user can create a library of "MACRO" files for any purpose where selecting channels via radio CAT control may be the most used purpose.



| Name | Date modified | Type |
|---|---|---|
| RGN_Net_Alpha_4Mhz_USB | 3/14/2016 4:14 PM | MAC File |
| RGN_Net_Bravo_5Mhz_USB | 3/14/2016 4:15 PM | MAC File |
| RGN_Net_Charlie_6Mhz_USB | 3/14/2016 4:15 PM | MAC File |
| RGN_Net_Delta_7Mhz_USB | 3/14/2016 4:16 PM | MAC File |
| RGN_Net_Echo_8Mhz_USB | 3/14/2016 4:17 PM | MAC File |
| RGN_Net_Foxtrot_9Mhz_USB | 3/14/2016 4:17 PM | MAC File |
| WWV_5Mhz_AM | 1/29/2014 9:35 PM | MAC File |
| WWV_10Mhz_AM | 1/29/2014 9:35 PM | MAC File |
| WWV_20Mhz_AM | 3/14/2016 4:13 PM | MAC File |

File name: *.MAC

MMI Macro files (*.MAC)

**NOTE:** The peer-to-peer message window as seen above is planned to function as an MMI Command Line in the future.

Here is an example of sending the Frequency of 10Mhz. to a Kenwood radio on VFO A:

```
radcmd FA00010000000;
```

Here is an example of sending both Frequency and Mode as one argument to a Kenwood radio on VFO A as one argument where the semi-colon termination is used for each command packet sent to the radio:

```
radcmd FA00010000000;MD2;
```

Here is an example of sending the same commands but as two arguments:

```
radcmd FA00010000000; MD2;
```

It is obvious that as Kenwood allows packing commands, as such the limit of 5 argument with RADCMD is rather moot.

HEXRADCMD supports sending HEX bytes to radios such as ICOM and old TEN-TEC models, older YAESU models and others. The process of using HEXRADCMD is similar to the use of RADCMD except that each byte of the radio command must be entered as a HEXRADCMD Argument, which requires a space character be inserted between each data byte of the radio command.

HEXRADCMD also supports the sending of command to radios that used a mixture of printable and non-printable characters. A good example of controlling a Cubic receiver for guard channel application in receiving broadcasts. The Cubic receiver requires one or more start bytes that are non-printable, followed by the receiver address and command and then a command terminator type. Here is an example of a mixed command set that sets the frequency on a Cubic R3030 receiver that requires three start bytes mixed with other characters would like the following when sent to the radio where the pound (#) sign represents the two characters that follows to be a HEX byte in a serial spy monitor tool:

```
#02#02#0201F01200000#0D
```

Broken down there are three HEX start bytes: #02#02#02

The R3030 set to ASCII address: 01

The ASCII frequency information is: F01200000

The radio command string termination is a carriage return as HEX byte: #0D

The same radio command packet represented completely in HEX would be:

```
#02#02#02#30#31#46#30#31#32#30#30#30#30#30#0D
```

To send this command to the R3030 using HEXRADCMD you would enter:

```
hexradcmd 02 02 02 30 31 46 30 31 32 30 30 30 30 30 0D
```

Here are some examples for ICOM and YAESU models that take into account the actual ICOM radio model HEX address, when using HEXRADCMD you can substitute 00 for the actual radio address you are directly sending these commands to the radio. Thus for the ICOM 746PRO examples below you would change the third byte which is the radio address from 66 to 00 for the address, 00 is the ICOM universal listener address to send the same to command to all devices on the bus if you were using more than one radio at a time.

Set ICOM 746PRO using factory address 66h to USB at 14.109Mhz:

```
hexradcmd FE FE 66 E0 05 00 90 10 14 00 FD FE FE 66 E0 06 01 FD
```

Set ICOM 746PRO using factory address 66h to USB at 14.109Mhz:

```
hexradcmd FE FE 66 E0 05 00 90 10 14 00 FD
```

Set ICOM 746PRO using factory address 66h to USB:

```
hexradcmd FE FE 66 E0 06 01 FD
```

Set ICOM 746PRO using factory address 66h to LSB:

```
hexradcmd FE FE 66 E0 06 00 FD
```

Set ICOM 746PRO using factory address 66h to USB-D:

```
hexradcmd FE FE 66 E0 06 01 FD FE FE 66 E0 1A 06 01 FD
```

Set ICOM 746PRO using factory address 66h to LSB-D:

```
hexradcmd FE FE 66 E0 06 00 FD FE FE 66 E0 1A 06 01 FD
```

Set ICOM 7200 using factory address 76h to USB-D: (NOTE: USB-D/LSB-D selection differs with 7200 vs. all previous ICOM models)

```
hexradcmd FE FE 76 E0 06 01 FD FE FE 76 E0 1A 04 01 01 FD
```

Set ICOM 7200 using factory address 76h to LSB-D:

```
hexradcmd FE FE 76 E0 06 00 FD FE FE 76 E0 1A 04 01 01 FD
```

Set Yaesu FT847 to USB at 14.109Mhz:

```
hexradcmd 01 00 00 00 07 01 41 09 00 01
```

Set Yaesu FT847 to 18.106Mhz:

```
hexradcmd 01 81 06 00 01
```

Set Yaesu FT847 to USB:

```
hexradcmd 01 00 00 00 07
```

Set Yaesu FT847 to LSB:

```
hexradcmd 01 00 00 01 07
```