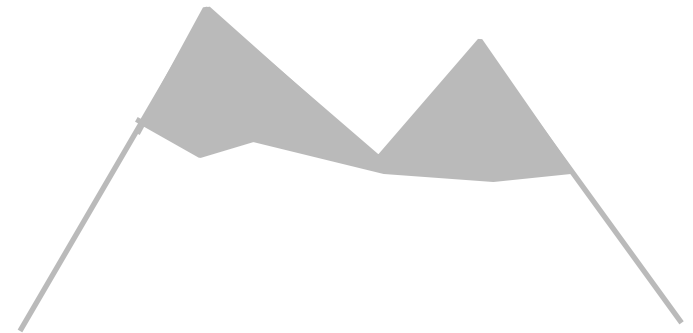


MicroFinder Doppler

Command Reference Guide



Version 1.0.3

April 14, 1998

© Copyright 1997-1998, AHHA! Solutions

All Rights Reserved

***MicroFinder* Doppler Command Reference**

Table of Contents

This document describes the commands available in the *MicroFinder* Doppler. We believe that the information contained herein, to the best of our ability, is accurate. If you believe you have found an inaccuracy in this manual we would be interested in hearing about it.

Introduction	2
Section 1 User Commands	3
Section 2 Advanced Command	9
Section 2 Debug Commands	15
Appendix A Button Function Codes	23
Appendix B Process ID Codes	23
Appendix C EEPROM Format	24
Appendix D CPU Memory Map	25

Introduction

This reference describes each command used by the *MicroFinder* Doppler. You can use this for information on a specific command. If you need information on how to operate or calibrate the doppler unit, please refer to the “*MicroFinder* Doppler Operator’s Guide.”

Commands are sorted into three sections.

- The first section, **User Commands**, contains commands that are most likely to be used in day-to-day operation of the doppler.
- The second section, **Advanced Commands**, is for commands that may be used to setup or “tune” the doppler for “out of the ordinary” type of operations.
- The third section, **Debug Commands**, may be used when constructing, debugging, or testing the doppler and its hardware.

Warning: Commands in the Advanced or Debug sections are normally not used for “day to day” operation of the doppler. These commands in some cases can cause you to lose your settings and may require a total reset of the MicroFinder system.

Each command is detailed on one page. The conventions used in this manual are as follows. Optional arguments for the command are shown in square brackets. This means you do not have to enter the

argument, but if you do enter the argument *do not* enter the brackets. An example of optional arguments is shown in the rotation command:

```
rotation [1-5] | [?]
```

When a choice of arguments possible, each alternative argument is shown separated by an “or bar”. The rotation example just used also shows an alternate argument and or-bar. Each command description includes examples of use. All examples of output from the doppler are printed in the Courier Font.

Each command must be spelled correctly, and there may be zero or more arguments. When the doppler can not understand your command, you will be informed with a response of “QSD!”. This is the doppler’s way of telling you the attempted command could not be executed.

Section 1

User Commands

User commands supply the most basic, and commonly used, functionality for MicroFinder configuration.

User Commands	
asw	Set or show the antenna configuration
baud	Set or show the RS-232 baud rate
bearing	Show the current doppler bearing
btap	Set or show the button table assignments
compass	Show the compass heading
eesave	Save configuration info to non-volatile memory
help	Print a short description of the MicroFinder commands.
mycall	Set your callsign in the unit
reset	Restart the MicroFinder

asw

asw [# 0|1]

asw is the antenna switch configuration command. Use this command when you wish to set or change the number of antenna elements or the logic switching level generated by the CPU. The doppler supports antenna configurations of 3, 4, 6, and 8 element antennas.

With no argument, **asw** will show the current configuration in memory. This may be different from the parameter stored in EEPROM if you have not yet saved your configuration.

Examples:

```
dop >asw
Antenna elements : 04
Antenna switch   : low
```

```
dop >asw 8 1
Antenna elements : 08
Antenna switch   : high
```

baud

baud [12|24|48|96]

The **baud** command is used to display or set the baud rate of the serial port. Each argument refers to the baud rates; 1200, 2400, 4800, or 9600.

With no arguments this command will show the current baud rate setting. When supplying an argument, the baud rate (if the configuration is save via **eesave**) will be used upon reset. This setting is not applied until the MicroFinder is reset.

The default baud rate is 4800. This value was selected because most GPS units provide serial data at 4800 baud.

Examples:

```
dop >baud 48
```

```
dop >baud
4800
```

bearing

bearing

The **bearing** command will display the current doppler bearing to the transmitter. The bearing information is of the form “%relative/quality%absolute”. If you have not configured and connected a compass for your doppler, there will be no absolute information. See the “opt” command for information on configuring your compass.

Examples:

```
dop> bearing
%91.8/5%111.6
```

btab

btab [B# F#] | [?]

The **btab** command is used to display or set functions associated with external buttons on doppler. The doppler supports up to 8 buttons. Buttons are numbered 1 through 8, and the functions are specified with a unique 4 character code. The button function codes are in Appendix A to this manual.

With no arguments, **btab** will show the current button configuration. The “?” argument will display a table of the button function codes. Setting a button function is accomplished by supplying button number and function code.

Examples:

```
dop >btab 5 bcf1
```

```
dop >btab
```

```
Button #1 : bpoo - Turn the pointer on and off
```

```
Button #2 : bptr - Switch the stepper motor speed
```

```
Button #3 : bsnp - Turn snooping on and off
```

```
...etc...
```

compass

compass

The **compass** command will display the current heading, as determined by the magnetic compass. If you do not have a compass configured and attached, this command is not useful. See the “opt” command for more information on configuring your compass.

Examples:

```
dop> compass  
91.8
```

eesave

eesave

The **eesave** command is used to save the configuration to the EEPROM. This command writes over the contents of the EEPROM, deleting your old configuration. **NOTE:** You can not undo this command.

Use this command after you have configured the doppler for your set up. Read the section “Configuring Your Doppler” in the User’s Guide.

Warning: If you received an EEPROM load error at start up time (see the Operator’s Guide, Error messages), do not save unless you are certain that all of the MicroFinder parameters are what you want.

help

help [u|a|d|!]

The **help** command is used to print a summary of the commands to your screen. Using no arguments, the normal user command summaries are printed. Help will display commands associated with “user” (u), “advanced” (a), “debug” (d), or all (!) commands.

The **help** command can also be run by using the “?” command.

mycall

mycall callsign

The **mycall** command is used to customize your doppler. When entered this way, your callsign will be used as the prompt when a computer is talking to your doppler. The parameter “callsign” should be a string containing your call, with any punctuation you wish. The command will actually take up to 8 characters and can be any printable characters.

Example

```
dop> mycall kn6zt ->  
kn6zt ->
```

This command is not included in the online help.

reset

reset

The **reset** command performs the same function as if you were to press the reset hardware button. This will restore the doppler to the “power on” state.

Section 2 Advanced Commands

These commands are for users wishing to utilize advanced features of the MicroFinder.

Advanced Commands	
aprs	Set or show the APRS parameters
calv	Start the calibration process
execbtn	Execute a button function (for host computer use)
opt	Enable or disable user supplied MicroFinder options.
ps	Display the state MicroFinder software processes.
qual	Show the current bearing quality value
rotation	Set or show the antenna rotation rate
sleep	Suspend the operation of a MicroFinder process
wake	Resume the operation of a MicroFinder process

aprs

aprs t|f [q [d]]

The **aprs** command is used to start a process to send periodic bearing and quality information through the serial line. The primary use of this is to provide a data stream compatible with APRS™.

The first argument turns the trace process on or off. Use an argument value of “t” or “1” to start the trace, and “0” or “f” to stop the trace.

The second argument is the *minimum* quality level to be reported. Any quality level, zero to nine is allowable. Quality zero will permit all bearings to be reported. Quality nine will only permit the best quality signals to be reported.

The third argument is the number of seconds between bearing reports. Reasonable delay values are one to sixty seconds, depending upon application. Legal values are zero to 3000 seconds (fifty minutes).

APRS bearing reporting is implemented in the *snoop process (psnp)*. This function will start the process. You can also turn this on via a button (button function “bapr”).

The format of the output is:

%ddd.d/q

or

%ddd.a/q%aaa.a

In each case, “ddd.d” is the relative bearing, in degrees. The quality of the bearing is represented by “q”. Quality information is on a scale of 0 to 9, 0 is the worst and 9 is the best.

If you have configured your system for a compass, the value “aaa.a” is the absolute bearing information in degrees.

Example:

```
dop >aprs 1 5 10
%91.8/5
%111.6/7
```

See the MicroFinder Construction Guide for information on building the correct cables for computer operation.

calv

calv [# ##] | -

The **calv** command is used to set or display the doppler calibration values. With no arguments, **calv** will display 5 calibration values corresponding to each of the 5 antenna rotation rates. Each calibration value is in the range of 0 to 199.

The command “**calv -**” will place the *MicroFinder* into “deferred” calibration mode. This mode is used by people who do not have a portable computer to put into their vehicle.

Examples:

```
dop> calv
 18 32 42 90 173
dop> calv 1 6
dop> calv
 6 32 42 90 173

dop> calv -
```

Hint: Write down the calibration values. If the EEPROM accidentally gets erased, you can reenter them without recalibrating.

execbtn

execbtn XXXX

The **execbtn** command is used to execute a button function, regardless of the current button table assignments. This allows you (or a computer program) to access all button functions.

The argument “XXXX” is a button function code. The function codes are specified in Appendix A of this manual.

Example:

```
dop >execbtn bswt
dop >
```

opt

opt [+c | -c | +q | -q]

The **opt** command is used to set or show the current option configurations. Version 1 of the *MicroFinder* firmware allows two options:

- magnetic compass
- quick look

Using arguments “+c” or “+q” will turn on the compass or the quick look options respectively. Arguments “-c” or “-q” will turn off the options.

Example:

```
dop >opt +c
+compass
-quicklook
dop >opt +q
+compass
+quicklook
dop >opt
+compass
+quicklook
```

ps

ps [t|1|f|0]

The **ps** command is used to show the process states. Each process in the doppler is either idle (sleeping) or running (awake). This command also lists the state of the driver processes. Supply an argument of “t” or “1” to see a description each process.

For more information on the processes in the doppler read the “Doppler Software Architecture” section in the Operator’s Guide.

Example:

```
dop >ps t
PROC STATE
pbut run User button processor
patt sleep Fancy LED display, IMPRESS
pdop run Display doppler data on LEDs
psnp sleep Send doppler data out serial line
pdil run Send doppler data to stepper
...etc...
```

qual

qual

The **qual** command displays the current measurement of the quality of the signal. This value is in the range of 0 to 9. A value of 0 is the worst, and 9 is the best.

Quality values correspond to the 7 segment quality display as follows:

<u>quality</u>	<u>bars</u>
1 - 3	1
4 - 6	2
7 - 9	3

Example:

```
dop >qual  
7  
dop >
```

rotation

rotation [1-5 |-]

The **rotation** command will set or show the rotation antenna rate. With no arguments, the current rate will be displayed. With a number argument 1 to 5, the corresponding rate will be set. With the “-” argument, a table showing the rate numbers and their corresponding rate, in Hertz, will be shown. The rates in Hertz are approximate values.

Read the section “Configuring Your Doppler” in the Operator’s Guide for more information on Rates.

sleep

sleep XXXX

The **sleep** command will cause the process specified by the process ID (XXXX) to become idle (sleeping). The process will not wake until you issue a **wake** command or some other function wakes the process. See Appendix B for the process ID codes.

Example:

```
dop >sleep psnp
```

wake

wake XXXX

The **wake** command will start a process that is currently idle. The process specified by its process ID (XXXX) will start running. Waking a currently running process has no effect. See Appendix B for more information on process ID values.

Example:

```
dop >wake psnp
```

Section 3 Debug Commands

These commands are used when building the MicroFinder, or when debugging problems.

Debug Commands	
ccw	Move the pointer display one step counter clockwise
cw	Move the pointer display one step clockwise
eedump	Display the raw contents of the non-volatile store
eer	Read one location of the non-volatile store
eew	Write one location of the non-volatile store
eezero	Erase the non-volatile store
rb	Read a byte from memory (RAM only)
rw	Read a word from memory (RAM only)
seg7	Write a value (0-9,A-F) to the 7 segment display
stepto	Move the pointer display to a specific location
time	Show the amount of time the MicroFinder has been running.
wb	Write a byte into memory (RAM only)
ww	Write a word into memory (RAM only)

CCW

ccw

The **ccw** command is used to move the stepper pointer one step in the counter-clockwise direction. This command is used to test the pointer motor wiring. Use this command only after you have done a **stepto** command. This sequence of commands will stop the pointer process. To re-enable the pointer, reset the doppler hardware, or wake the pointer process (pdil).

Example:

```
dop >stepto 62
```

```
dop >ccw
```

```
62
```

CW

cw

The **cw** command is used to move the stepper pointer one step in the clockwise direction. This command is used to test the pointer motor wiring. Use this command only after you have done a **stepto** command. This sequence of commands will stop the pointer process. To re-enable the pointer, reset the doppler hardware, or wake the pointer process (pdil).

Example:

```
dop >stepto 62
```

```
dop >cw
```

```
63
```

```
dop >ccw
```

```
62
```

eedump

eedump

The **eedump** command will display the raw contents of the EEPROM. Appendix C details the layout of the EEPROM.

If you want to change a configuration parameter without using a regular user command you can do so with the **ew** command. However, this is not a recommended approach for changing configurations.

Example:

Print out the EEPROM to screen

```
dop >eedump
00 - 0003 0340 0603 0501 0000 0204 0000 0000
08 - 0000 0000 0101 0001 0001 0101 0100 0000
10 - 0000 0000 0000 0000 0000 0000 0000 0000
18 - 0000 0000 0000 0000 0000 0000 0000 0000
...etc...
```

eer

eer ##

The **eer** command will read one EEPROM register at a time. You can get most of the same information from the **eedump** command. Output from this command is in hexadecimal.

Example:

```
dop >eer 01
034f

dop >eer 02
c603
```

eeW

```
eeW AA ####
```

The **eeW** command will write one EEPROM register at a time. You can use this to patch locations in the EEPROM. Unless you really know what you are doing, beware! This command can really mess with your mind, and your doppler. See Appendix C for the layout of the EEPROM.

Example:

```
dop >eeW 01 0340
```

```
a7e0
```

```
dop >eer 02 0603
```

```
12fe
```

eezero

```
eezero
```

The **eezero** command is used to clear the EEPROM to an initial, empty state. You will not need to use this command often, if at all. Use this when you believe the data in the EEPROM has become corrupted. The doppler will then reset to “factory defaults” which can then be saved using the **eesave** command.

rb

rb AAAA

The **rb** allows you to read a single byte from the CPU memory. The address value is anything between 0x0000 and 0xffff. For specific locations in memory, see Appendix D. The argument AAAA and output of this command are in hexadecimal.

Example:

```
dop >rb fd8c
42
```

rw

rw AAAA

The **rw** allows you to read a single word from the CPU memory. The address value is anything between 0x0000 and 0xffff. For specific locations in memory, see Appendix D. The argument AAAA and output of this command are in hexadecimal.

Example:

```
dop >rw fd8c
42ef
```

seg7

seg7 [#]

The **seg7** command is used to place a hex digit value in the seven segment display. Legal values are 0-9 and A-F. The main use for this command is to test the wiring on the 7 segment display, or sometimes for a host computer to display a value.

Example:

```
dop >seg7 4
```

stepto

stepto ##

The **stepto** command will cause the the connection between the doppler and the stepper pointer to become disconnected. The value specified will become the target for the pointer. You can then use the **cw** and the **ccw** commands to move the pointer in one direction or another. These commands are mainly used to debug wiring of the pointer stepper motor. Valid target values are 0 to 199 (“doppler degrees”).

WARNING: This command will stop the pointer process, “pdil”. You can restart the process with the “wake” command, or by resetting the doppler.

Example:

```
dop >stepto 62
```

```
dop >cw  
63
```

time

time

The **time** command shows the amount of elapsed time since the last doppler reset or power up.

The output is hours, minute, and seconds (hh:mm:ss). The realtime clock in the doppler can record about 32,000 hours. This is almost 4 years elapsed time. We hope this this enough!

Example:

```
dop >time  
48:23:17
```

wb

wb AAAA ##

The **wb** command is used to stuff a byte value into a location in memory. This command can be used to patch some runtime parameters or write values to certain output ports. This is normally only used when debugging a newly constructed doppler. The arguments AAAA, ## and output of this command are in hexadecimal.

Example:

```
dop >wb fdfc 42
```

WW

`ww AAAA ####`

The `ww` command is used to write a word value into memory. This is used to patch some runtime values. Normally this command is only used when debugging a newly constructed doppler. This command writes each byte with a separate CPU instruction. See Appendix D for a list of important locations in memory. The arguments `AAAA`, `####` and output of this command are in hexadecimal.

Example:

```
dop >ww fdfc 1235
```

Appendix A

Button Function Codes

bapr Send bearing to APRS
batt Toggle LED attract display
bacc Save calibration values
bcal Set calibration values
bcmp Start compass calibrate mode
bswt LED display - relative, absolute,
compass
bptr Pointer motor speed
bpoo Turn the pointer on and off
brat Change antenna rotation rate
bsnp Turn snooping on and off
bhld Pause the LED display
bflt Turn on or off quality or quality+qfilt
bcf1 Print 'CB1' to the serial port
bcf2 Print 'CB2' to the serial port
bcf3 Print 'CB3' to the serial port
bcf4 Print 'CB4' to the serial port

Appendix B

Process ID Codes

pbut Button input
patt "Attract mode", for fun only.
pled Display doppler or compass
psnp Send doppler data out serial line
pdil Doppler pointer
danl Analog inputs.
dser RS-232 input
dstp Stepper motor driver
dled Manage LED display
dcal Calibration driver
dcmp Compass calibration driver

Appendix C EEPROM Memory Map

The memory map of the EEPROM is presented for informational purposes. AHHA! Solutions does not support user modification of the contents of the EEPROM.

Address	Description
\$00	EEPROM Version
\$01	Antenna Rate
\$02	Button Functions #1 & #2
\$03	Button Functions #3 & #4
\$04	Button Functions #5 & #6
\$05	Button Functions #7 & #8
\$06	Deferred Calibration Functions
\$07	Calibration Values
\$08	Calibration Values
\$09	Calibration Values
\$0A	Compass Calibration Values
\$0B	Compass Calibration Values

Address Description	
\$0C	Compass Calibration Values
\$0D	Compass Calibration Values
\$0E	Filter Parameters
\$0F	Serial Port Configuration
\$10	Option Parameters
\$11	Pointer Parameters
\$12	Filter Parameters
\$13	Filter Parameters
\$14	APRS Command Parameters
\$15 - \$19	MyCall String Data
\$1A - \$21	Process Table

Appendix D

CPU RAM Map

The memory map of selected MicroFinder RAM locations is presented for informational purposes.

AHHA! Solutions does not support user modification of the contents of the RAM.

Address	Description
\$FDA8	Clock hour
\$FDAA	Clock minutes
\$FDAB	Clock seconds
\$FDAC	Clock milliseconds
\$FDAE	Clock micro seconds
\$FDB4	Process status serial input
\$FDB5	Process status button input
\$FDB6	Process status "attract mode"
\$FDB7	Process status LED Doppler display
\$FDB8	Process status APRS snooper
\$FDB9	Process status analog input
\$FDDBA	Process status pointer doppler display
\$FDBB	Process status pointer driver

Address	Description
\$FDBC	Process status LED driver
\$FDBD	Process status calibration driver
\$FDBE	Process status compass calibrate driver
\$FDC2	Button 1 function
\$FDC3	Button 2 function
\$FDC4	Button 3 function
\$FDC5	Button 4 function
\$FDC6	Button 5 function
\$FDC7	Button 6 function
\$FDC8	Button 7 function
\$FDC9	Button 8 function
\$FDD0	Print string (CString)