Canon EF-232 Library
User Manual

BIRGER
engineering

Manual Rev 1.3
10/16/09

This manual covers library versions up to 20c and bootloader versions up to 45.

Birger Engineering, Inc.
38 Chauncy St #1101
Boston, MA 02111
http://www.birger.com

1   Table of Contents

2   Quick Reference Tables

Table 1. Library Command Quick Reference

Command	Arguments	Description
bv		Print the bootloader version.
da		Print aperture information.
de		Dump EEPROM.
ds		Print distance stops.
dz		Print the zoom range.
eh	pos,checksum	Set absolute lens focus position (0…0x3FFF).
ex		Exit to the bootloader.
fa	position	Move focus to absolute position.
fd		Print focus distance range.
ff	position	Fast focus.
fp		Print the raw focus positions.
gs		Echo current device and lens statuses.
hv		Print the hardware version.
id		Print basic lens identification (zoom and f-number).
in		Initialize the aperture motor. Aperture will fully open.
is	{0, 1}	Turn image stabilization off/on.
la	options	Learn the focus range.
lc		Print cached lens status
ll		Library loaded check.
lp		Lens presence.
ls		Query lens for status immediately and print.
lv		Print the library version string.
ma	stop	Move aperture to absolute position.
mc		Move aperture to the fully stopped down limit.
mf		Move focus incremental.
mi		Move focus to the infinity stop.
mn	num_stops	Move aperture incremental.
mo		Move aperture to completely open.
mz		Move focus to the zero stop.
pa		Print the aperture position.
pf		Print the focus position.
pl	{0, 1}	Lens power
rm	verbose[,new]	Set response modes.
se	byte,val	Temporarily set non-volatile (EEPROM) byte.
sf	count	Set the focus counter.
sg	port,state	Set GPIO.
sm	mode_flags	Set special modes.
sn		Print the device serial number.

sr	{0, 1}	Set spontaneous responses off/on.
vs		Print the short version string.
we	serial_number	Write non-volatile parameters to EEPROM.

Table 2. Bootloader Command Quick Reference

Command	Arguments	Description
be		Print the banner string.
bl		Boot library.
br	baud_rate[,prt]	Set baud rate, argument is actual baud rate / 100.
bs		Obsolete – commit baud rate setting.
hv		Print the hardware version.
lv		Print version of loaded library.
qp		Query current port number.
rm	verbose[,new[,prt]]	Set response modes.
routeesc,	depth	Escape from port routing.
rt	port,depth	Route through to port with assigned depth.
sn		Print the unit serial number.
sp	{0, 1}	Suppress prompt.
vn		Print the long version string.
vs		Print the short version string.
wc		Write all configurations to EEPROM.
xm	unlock_sn	XModem firmware download.

3   General Notes on Using the Library

This section describes the differences between legacy library commands (libraries up to version 15) and new library commands (versions 16 and upward), the differences in command behavior, and the differences between the bootloader and library. It is also the reference for the basic behaviors of the library.

3.1   Bootloader vs. Library

Prior to version 16 of the library, the bootloader and library code were integrated together. While the library was still separately upgradeable, the bootloader and library worked together. All commands were available for access all of the time. All commands blocked until they were finished.

Version 4 of the hardware shipped with a new bootloader and new command model. Bootloader commands are only available from within the bootloader, and library commands are only available from within the library, except where the functionality is duplicated. If a command with the same name exists in the bootloader and library then it will perform the same function in both instances.

The bootloader transfers execution to the library for normal operation, and the library handles normal command processing. Commands can be blocking or non-blocking (they return immediately while lens processing occurs "in the background"). Firmware upgrading can only be done through the bootloader.

3.2   Aperture Positioning

Aperture positioning is generally measured in ¼ stop increments. However, the input to the aperture positioning commands is an integer and the reported aperture positions are f-number times ten or a corresponding integer number of ¼ stops. For example, if f28 is reported for an aperture position, then it is indicating that the actual f-number is 2.8. If this is the wide-open position then it is considered the zero position, and the input to the absolute aperture position commands would be zero. If aperture measurement is not reported as ¼ stops then the exception will be noted where applicable.

A value of one would indicate ¼ stop from the full-open position (2.8) towards the closed position, which would be approximately 3.1. If the aperture position is then reported in this state, the number one would be reported as the position in number of ¼ stops from the wide-open (zero) position, and the f-number would be reported as f30. Note that the library uses integer rather than floating point math, so the f-number reported is close to, but not exactly the mathematical f-number.

3.3   Legacy Focus Positioning

Focus positioning for legacy commands is performed with respect to the encoder positions directly reported by the connected lens. Each lens will have a different encoder count for its focusable range, and the encoder count for the zero position will almost

never be zero, in fact it will usually be random, changing with each power cycle of the lens. However, the focus counter command 'sf' can be used to make any number the reference for the zero position. The lens can be focused anywhere from zero to infinity by using commands to learn the full encoder count range and then applying those values to the positioning commands. For example, a move to zero and then a set focus counter to zero will establish the value of zero as the position for the minimum focus distance of the lens. A move to infinity will then report the full number of encoder counts that it took to get there.

3.4   Mapped Absolute Focus Positioning

Library versions 16 and up introduce the ability to perform absolute positioning on a fixed scale. This fixed scale is a 14-bit range from 0 to 16383 (hex 3FFF). The range of the lens from the zero stop to the infinity stop is mapped into this numerical scale. The 14-bit range is generally much larger than the encoder range returned by the lens, and therefore a small change in the mapped position often will not result in a change to the focus position.

The mapped focusing feature is highly recommended when using a control surface to alter focusing. It allows use of a scale that is always fixed between the zero and infinity positions. However, if the control surface is not capable of sending small enough changes in the mapped position, then jumpy focusing behavior can occur. Ideally the control surface should also have a 14-bit range, but a smaller range is okay as long as it is several times larger than the lens' encoder range. The control surface should have at least 12-bit resolution to meet this requirement for all lenses tested to date. A smaller resolution will still work, but focusing will not be as smooth or precise.

3.5   Focus Drift

Most lenses will experience drift of the focus mechanism. This results in a change of the encoder position associated with a particular stop. Generally the lens will lose or gain a few encoder counts when hitting a stop. The lens may also lose positioning just during normal focusing, and it can be made worse if the focus move is small. The better-constructed, more-expensive lenses will drift far less than the cheaper lenses. Drift must be accounted for by relearning the encoder positions of the zero and infinity stops.

3.6   Legacy vs. New Commands

The library as shipped emulates the legacy command protocol. In this protocol all commands are case insensitive and are acknowledged with the string 'OK' to indicate that the command was received. This is followed by any output that the command may have, after any delay that the command may need to complete its processing. Commands that are marked as 'new' in this manual will still conform to this behavior.

It is possible to change this behavior so that the 'OK' string is not emitted. In this case commands will report only their output, or a confirmation if they have no regular output. These modes can be configured with the response mode command, and are

described further under that command heading. When the new library protocol is active, commands become case-sensitive. They must be entered as lower-case only.

There are general enhancements that apply regardless of mode or command. White space is now permitted and commands can have multiple arguments, each separated by a comma.

3.7   Blocking vs. Non-Blocking Commands

In the legacy libraries (versions 15 and lower), all commands were blocking. In other words, they needed to finish before any other command could be sent. The new libraries introduce two new servo routines: an aperture servo and a focus servo. There are new commands to use these routines. These new commands return immediately and the library handles the necessary lens processing in the background.

If you intend to have multiple devices talking to the lens controller over several serial ports, then all lens commands and status queries must use the new commands that operate in the background. Otherwise, a device on one port will hold off the other devices from getting their communications through in a timely manner. Also, the library automatically coordinates the lens processing so no conflicts occur, which would not be possible if using the legacy commands from multiple serial ports.

3.8   Terse vs. Verbose Reporting Modes

The device can be configured to report in two different modes in addition to legacy/new protocols using the response modes ('rm') command. In terse mode, characters are not echoed back from the device, and all commands return a single number, zero for no error or a non-zero error value. Note that even commands that would normally return a status string will not do so in terse mode as long as they are marked as legacy commands. New commands will always echo their output if they have status to print.

In verbose mode, all characters input to the device are echoed back out, and commands will print their full status strings as usual. If set to both verbose and new protocol modes, then a question mark will be emitted by the device for unknown commands. Verbose mode is most useful when manually controlling the device through a terminal program.

3.9   Status Strings

Starting with library version 18, there are a standard set of status strings that can be queried for or spontaneously emitted from the device. The spontaneous emissions are off by default but can be turned on with the command 'sr'.

3.9.1   Focus Position Status

Format: %:xxxx

Where xxxx is a 4-digit hexadecimal number indicating the current focus position of the lens in the mapped range (0...3FFF).

3.9.2   Aperture Position Status

Format: &:xx

Where xx is a 2-digit hexadecimal number indicating the current aperture position in 1/8 stops. Note that if the aperture position is unknown, for example if a lens has been connected but not initialized, then the status string will be:

&:?

3.9.3   Focal Length and Aperture Range Status

Format: @:<focal_length>mm,f<min_f>,<num_stops>,f<max_f>

Where <focal_length> is a decimal number for the current focal length in millimeters, <min_f> is a decimal number for the current minimum f-number times ten, <num_stops> is a decimal number for the number of ¼ stops available, and <max_f> is a decimal number for the current maximum f-number times ten.

3.9.4   Miscellaneous Status Flags

Format: #:xxxx

Where xxxx is a 4-digit hexadecimal number indicating the current value of the flags. The flags are given in the following table.

Table 3. Status Flags

Bit(s)	Description
0	Port RA0 is input; 0 = output, 1 = input.
1	Port RA1 is input; 0 = output, 1 = input.
2	Port RA0 value; 0 = low, 1 = high.
3	Port RA1 value; 0 = low, 1 = high.
4	Aperture initialized; 0 = not initialized, 1 = initialized.
5	Manual focus; 0 = lens is in autofocus, 1 = lens in in manual focus.
6	Focus mode; 0 = normal; 1 = sticky lenses
7-15	Reserved.

3.9.5   Lens Connection Messages

When in new-protocol mode, a spontaneous string will be emitted when a lens is connected or disconnected from the unit. These strings are:

Lens Connected.

and

Lens Disconnected.

3.10   Errors in Command Processing

If a command fails it will report a numerical error code. In terse mode the number is returned by itself, while in verbose mode and new-protocol mode the return string will be:

ERRx

Where x is the numerical error code. The possible error codes that can be returned are given in the following table.

Table 4. Error Codes

Error Code	Description
0	No error.
1	Unrecognized command.
2	Lens is in manual focus mode.
3	No lens connected.
4	Lens distance stop error.
5	Aperture not initialized.
6	Invalid baud rate specified.
7	Reserved.
8	Reserved.
9	A bad parameter was supplied to the command.
10	XModem timeout.
11	XModem error.
12	XModem unlock code incorrect.
13	Not used.
14	Invalid port.
15	License unlock failure.
16	Invalid license file.
17	Invalid library file.
18	Reserved.
19	Reserved.
20	Not used.
21	Library not ready for lens communications.
22	Library not ready for commands.
23	Command not licensed.
24	Invalid focus range in memory. Try relearning the range.
25	Distance stops are not supported by the lens.

4   How to Read the Command References

Each command page is organized as follows:

<div style="border: 1px solid black; display: inline-block; padding: 10px;">

**Applicable Library
Versions**

</div>

Command Name (command)

Command Type	**: New/Legacy or both**
Syntax	**: What is entered as the complete command**
Returns	**: The information or exact string returned**

Description

A description of how the command is used.

Examples:

One or more examples of how the command is used. The command syntax is shown as normal text, information returned from the device is shown in italic text. The command confirmation for legacy behavior ('OK') is not shown.

5   Library Command Reference

The following pages contain descriptions of the commands that are available from within the library. These commands will not be active while in the bootloader, however the bootloader may have commands with the same name also implemented, in which case they will perform the same function.

5.1 Bootloader Version (bv)

Command Type : New
Syntax : bv

Returns : Bootloader version number

Description

      This command will print the version of the bootloader code from within the library. Therefore exiting to the bootloader is not needed to read the revision information.

Example:

```
bv <CR>
```

44

5.2 Define Aperture (da)

Command Type : **Legacy**
Syntax : **da**

Returns : **f<min>,<num_stops>,f<max>**

Description

This command will print the the aperture range of the connected lens. The range is printed in three parts as follows:

fmin**,**num_stops**,f**max

The item printed is the letter 'f' followed by the current minimum f-number. This is followed by a comma and the total number of stops in ¼ stops that can be commanded to the lens. This is then followed by another comma, the letter 'f', and the current maximum f-number.

Example:

```
da <CR>

f28,28,f320
```

5.3 Dump EEPROM (de)

Command Type : New
Syntax : de

Returns : **EEPROM parameters as hexadecimal bytes**

Description

This command will dump a hexadecimal stream of bytes that corresponds to the non-volatile parameters stored in EEPROM. The commonly configured parameters are almost always available through other commands, so this command is used to gain access to the remaining parameters. These parameters are often low-level timing configurations. Contact Birger Engineering if you think your programming needs might require modification to these parameters.

5.4 Distance Stops (ds)

Command Type : New
Syntax : ds

Returns : $:<far>:<near>:<encoder>

Description

This command will execute a move to the zero stop of the lens. It will then advance the lens through its focus range and print the distance stops as reported by the lens. The distance stops take the form:

$:<far>:<near>:<encoder>

Where <far> is a 4-digit hexadecimal number representing the far distance of the stop in centimeters, <near> is another 4-digit hexadecimal number representing the near distance of the stop in centimeters, and <encoder> is a 4-digit hexadecimal number of the raw encoder count of the lens. The command finishes with a move to the focus motor infinity stop to read its position followed by a return to the original focus position. Issuing this command while it is already in progress will cancel further distance reporting. Other lens control commands will not work or should not be issued while this command is in progress.

The infinity position is reported as ffff hexadecimal. The infinity position reported as a distance stop may not actually correspond with the infinity focus position resulting from the focus motor's infinity stop. These are often two different stops in the lens construction. Not all lenses are capable of reporting distance stops.

Example:

```
ds <CR>

$:0021:0020:fe01
$:0023:0021:feb4
$:0025:0023:ff23
$:0026:0025:ffb1
$:0029:0026:0066
$:0035:0029:0216
$:005f:0035:0436
$:008b:005f:04f6
$:017a:008b:05e8
$:045d:017a:062f
$:ffff:045d:0655
$:ffff:ffff:06bf
$:DONE
```

5.5 Define Zoom (dz)

Command Type : **Legacy**
Syntax : **dz**

Returns : **<min_focal_length>mm,<max_focal_length>mm**

Description

This command will print the zoom range of the lens. The first number printed is the minimum focal length in millimeters followed by 'mm'. This is followed by a comma, the maximum focal length of the lens, and the letters 'mm'. Prime lenses will print the same value for minimum and maximum focal length.

Example:

```
dz <CR>

18mm,55mm

dz <CR>

50mm,50mm
```

5.6   Servo Focus with Checksum (eh)

Command Type   : New
Syntax   : eh <absolute_position>,<checksum>

Returns   : Nothing

Description

This command is used to set the focus position of the lens on an absolute 14-bit scale from 0 to 3FFF hexadecimal. The command requires a checksum that is the bit-wise exclusive-OR of the four 4-bit digits of the position. This command executes through the servo focus routine and returns immediately.

All four digits of the position must be specified, including leading zeros. If the correct number of digits is not sent or the checksum is incorrect, then the command will be ignored.

Examples:

```
eh0000,0 <CR>

eh1000,1 <CR>

eh3fff,c <CR>
```

5.7 Exit to Bootloader (ex)

Command Type : New
Syntax : ex

Returns : In bootloader!

Description

This command exits library processing and hands control to the bootloader. Firmware (library) upgrades must be performed through the bootloader. There are several other commands that can only be accessed through the bootloader.

The return response from this command depends on which protocol mode is active, legacy vs. new, and terse vs. verbose, but the string 'In bootloader!' will always be returned.

Examples:

```
ex <CR>

r!
In bootloader!

ex <CR>

BEI Boot Executive, v44
(c)2008 Birger Engineering, Inc.
In bootloader!
```

5.8   Focus Absolute (fa)

Command Type   : Legacy
Syntax   : fa <position>

Returns   : DONE<rposition>,<flag>

Description

This command makes the focus mechanism move to the specified focus count position, specified in counts moved in <position>. If an input value would move the focus out of the legal range the value is rounded to the boundary (i.e. min/max count). <rposition> is the actual position that the count has been moved to, <flag> is 1 if the lens reports having hit a stop, and 0 if it hasn't. Note that some lenses do not return a 1 until the second time the stop is hit.

Examples:

```
fa100 <CR>

DONE100,0

fa2000 <CR>

DONE1759,1
```

5.9   Focus Distance (fd)

Command Type   **: Legacy**
Syntax   **: fd**

Returns   **: <near>cm,<far>cm**

Description

The number is returned as two distances, <near> the near distance, and <far> the far distance, both in centimeters, where 65535 represents infinity. Not all lenses return distance information. This information corresponds to what the lens reports as the stops that bracket the current focus position, not as the near and far focal distances as determined by the actual optics involved, which are calculations based on the lens focal length, aperture, focusing distance, and circle-of-confusion.

Examples:

```
fd <CR>
```

45cm,48cm

```
fd <CR>
```

65535cm,65535cm

5.10 Fast Focus (ff)

Command Type : Legacy/New
Syntax : ff <position>

Returns : DONE

Description

This command has been updated to use the servo focus routine. The position supplied uses the legacy focus positioning, which are encoder counts as reported by the lens with the applied offset set through the focus counter command 'sf'.

Example:

```
ff1000 <CR>
```

DONE

5.11   Focus Positions (fp)

Command Type   : New
Syntax   : fp

Returns   : fmin:<fmin> fmax:<fmax> current:<current_pos>

Description

This command will print the focus positions in raw encoder counts as reported by the lens. The positions are reported as signed decimals, and as they are raw encoder counts, do not have the focus counter offset applied. <fmin> is the zero position of the lens and is updated every time the lens is commanded to the zero stop with the 'mz' command. <fmax> is the infinity position of the lens and is updated every time the lens is commanded to the infinity position with the 'mi' command. Both positions are updated if the learn focus range ('la') command is issued. <current_pos> is the current focus position of the lens.

Example:

```
fp <CR>

fmin:-1303  fmax:1228  current:696
```

5.12   Get Statuses (gs)

Command Type   : New
Syntax   : gs

Returns   : Status strings

Description

This command prints complete status information in the form of the new status strings. The status strings are described in <u>Section 3.9</u> of this manual.

Example:

```
gs <CR>

%:1948
&:0a
@:100mm,f28,28,f320
#:0010
```

5.13 Hardware Version (hv)

Command Type   : Legacy/New
Syntax   : hv

Returns   : Device hardware version

Description

This command will print the hardware version of the device. It exists up to library version 15 in the unified bootloader/library. The command does not appear in versions 16 and 17 of the library, so obtaining the hardware version requires exiting to the bootloader for those versions.

Example:

hv <CR>

4

5.14   Lens Identification (id)

Command Type   : Legacy
Syntax   : id

Returns   : <current_focal_length>mm,f<current_min_f_number>

Description

This command prints the basic lens identity. For a zoom lens it will print the current focal length, otherwise it prints the fixed focal length of the lens. This is followed by a comma and the letter 'f' followed by the current minimum f-number that the lens is capable of multiplied by ten.

Examples:

```
id <CR>
```

38mm,f47

```
id <CR>
```

100mm,f28

5.15 Initialize Aperture (in)

Command Type : Legacy
Syntax : in

Returns : DONE

Description

This command initializes the aperture motor and position. This opens up the aperture to its minimum f-number (maximum opening). This position is then set as the zero position for all subsequent aperture movement commands. This is required before any of the aperture movement commands can be used, as driving the aperture motor too far without knowing its position could damage the aperture mechanism. The library prevents you from issuing aperture movement commands unless the position has first been initialized with this command.

Example:

```
in <CR>
```

DONE

5.16 Image Stabilization (is)

Command Type   : **Legacy/New**
Syntax   : **is {0, 1}**

Returns   : **DONE**

Description

Legacy: Up to version 17 of the library, this command turns on the image stabilization function of the lens. This command blocks until any character is input into the device or power is removed, so no other commands can be issued while it is active.

New: Version 18 of the library introduces the ability to turn this function on or off as a background process. In most cases lenses with image stabilization will already have the function active without needing this command. In that case, image stabilization should be turned on or off using the switch on the lens barrel. Lenses that use full-time manual focus, such as the Canon EF 85mm f/1.2 L II USM need to have this turned on or they will not be able to be focused manually.

Examples:

```
is0 <CR>
```

DONE

```
is1 <CR>
```

DONE

5.17 Learn Absolute Focus Range / Detect Lens (la)

Command Type   : New
Syntax   : la [<options{0,1,2}>]

Returns   : DONE:LA

Description

This command learns the focus range of the lens for the servo focus routine and also for any legacy commands in a single step. No other lens movement commands can be issued while this command is in progress. This command does not block, so other commands that don't require lens communications can be issued while the lens is focusing. A confirmation will be emitted by the device when the lens has completed focusing. The servo focusing and mapped focus range will not function unless the lens focus range is learned.

This command can also be supplied with an argument that starts the lens detection process at different stages.

Option: 0   Same as LA without arguments – learn the lens' focal range.
            1   Resynchronize with the lens.
            2   Re-detect and resynchronize with the lens.

Examples:

```
la <CR>

DONE:LA

la1 <CR>

DONE:LA

la2 <CR>

DONE:LA
```

5.18   Cached Extended Lens Status (lc)

Command Type   : New
Syntax   : lc

Returns   : @:<focal_length>mm,f<min_f>,<num_stops>,f<max_f>

Description

This command will print the extended lens status from cached information. The status string is of the form outlined in Section 3.9. Cached information is updated periodically by the library if background querying is turned on (see command 'sm'). If multiple devices are interfacing to the lens controller over multiple serial ports, then this command should be used to retrieve status, as it will avoid conflicts.

Example:

lc <CR>

@:38mm,f47,21,f293

5.19 Library Loaded (ll)

Command Type   : New
Syntax   : ll

Returns   : OK

Description

      This command simply prints 'OK'. It can be used as a test to see if the device is executing library code or bootloader code. The command does not exist in the bootloader, and therefore will not work unless library code is being executed.

Example:

```
ll <CR>
```

OK

5.20 Lens Presence (lp)

Command Type   : New
Syntax   : lp

Returns   : 0 or 1

Description

This command will indicate if a lens is currently connected or not. It prints the number '0' when no lens is connected and '1' when a lens is connected.

Examples:

```
lp <CR>
```

0

```
lp <CR>
```

1

5.21   Immediate Extended Lens Status (ls)

Command Type   : New
Syntax   : ls

Returns   : @:<focal_length>mm,f<min_f>,<num_stops>,f<max_f>

Description

This command will print the extended lens status by immediately querying the lens. The status string is of the form outlined in Section 3.9. This command should be used if background querying is turned off (see command 'sm').

Example:

```
ls <CR>

@:38mm,f47,21,f293
```

5.22   Library Version (lv)

Command Type : Legacy
Syntax : lv

Returns : **Full library identity string**

Description

This command prints the full library identification string.

Example:

```
lv <CR>
```

Canon EF-232 Library v17

5.23   Move Aperture Absolute (ma)

Command Type   : Legacy
Syntax   : ma <pos>

Returns   : DONE<rpos>,f<f_number>

Description

This command moves the aperture mechanism to the specified encoder position, specified in ¼ stops from the fully-open position, indicated by the user in <pos>. If an input value would move the aperture out of the legal range the value is set to the boundary (i.e. min/max aperture). <rpos> is the actual position the aperture moved to in ¼ stops from the fully-open position, and <f_number> is the absolute position given as the lens f-number times ten.

Examples:

```
ma0 <CR>

DONE0,f35

ma5 <CR>

DONE5,f54
```

5.24   Move Aperture to Closed Position (mc)

Command Type   : **Legacy**
Syntax   : **mc**

Returns   : **DONE<signed_num_steps>,f<f_number>**

Description

This command will move the aperture to the fully closed position. The number of steps in ¼ stops that the aperture actually moved is given in <signed_num_steps>. The final position is given in <f_number> as the lens f-number times ten.

Example:

```
mc <CR>

DONE14,f216
```

5.25   Move Focus Incremental (mf)

Command Type   : Legacy
Syntax   : mf <signed_counts>

Returns   : DONE<signed_num_counts>,<flag>

Description

This command makes the focus mechanism move incrementally the number of counts specified in <signed_counts>. If the number specified is positive then the focus moves towards the infinity position. If the number is negative then the focus moves towards the zero position. If the number of counts specified is out of range, the mechanism will move as far as it can, <signed_num_counts> is the actual number of counts moved, and <flag> is 1 if the lens reports having hit a stop, 0 if it hasn't. Note that some lenses do not return a 1 until the second time the stop is hit.

Examples:

```
mf100 <CR>

DONE100,0

mf-50 <CR>

DONE-50,0
```

5.26   Move Focus to Infinity (mi)

Command Type   : **Legacy**
Syntax   : **mi**

Returns   : **DONE<signed_num_counts>,<flag>**

Description

     This command moves the lens focus to the infinity position. The actual number of counts moved as reported by the lens' encoder is given in <signed_num_counts>. <flag> is 1 if the lens reports having hit a stop, 0 if it hasn't. Note that some lenses do not return a 1 until the second time the stop is hit.

Example:

```
mi <CR>

DONE246,1
```

5.27 Move Aperture Incremental (mn)

Command Type   : Legacy
Syntax   : mn <signed_steps>

Returns   : DONE<num_signed_steps>,f<f_number>

Description

This command moves aperture mechanism the specified number of steps, in the specified direction, in ¼ stops given by <signed_steps>. A positive step is a move towards the fully closed position and a negative step is a move towards the fully opened position. If an input value would move the aperture out of the legal range the value is clipped to the boundary (i.e. min/max aperture). <num_signed_steps> is the actual number of steps the aperture was moved, and <f_number> is the absolute position of the aperture after executing the move, in the lens f-number times ten.

Examples:

```
mn2 <CR>
```

DONE2,f54

```
mn-1 <CR>
```

DONE-1,f49

5.28 Move Aperture to Opened Position (mo)

Command Type : Legacy
Syntax : mo

Returns : DONE<signed_num_steps>,f<f_number>

Description

This command will move the aperture to the fully open position. The number of steps in ¼ stops that the aperture actually moved is given in <signed_num_steps>. The final position is given in <f_number> as the lens f-number times ten.

Example:

```
mo <CR>

DONE-21,f35
```

5.29   Move Focus to Zero (mz)

Command Type   : **Legacy**
Syntax   : **mz**

Returns   : **DONE\<signed_num_counts\>,\<flag\>**

Description

This command moves the lens focus to the zero position. The actual number of counts moved as reported by the lens' encoder is given in \<signed_num_counts\>. \<flag\> is 1 if the lens reports having hit a stop, 0 if it hasn't. Note that some lenses do not return a 1 until the second time the stop is hit.

Example:

```
mz <CR>

DONE-1246,1
```

5.30   Print Aperture Position (pa)

Command Type   : Legacy
Syntax   : pa

Returns   : <pos>,f<f_number>

Description

This command prints the current position of the aperture. <pos> is the absolute position given in ¼ stops from the fully-open position and <f_number> is the absolute position given as the lens f-number times ten.

Example:

```
pa <CR>
```

5,f54

5.31   Print Focus Position (pf)

Command Type   : **Legacy**
Syntax   : **pf**

Returns   : **Signed focus position in encoder counts.**

Description

This command prints the current position of the focus mechanism given in encoder counts with the user's desired focus counter offset applied.

Examples:

```
pf <CR>

-1247

pf <CR>

743
```

5.32   Lens Power (pl)

Command Type : **Legacy**
Syntax : **pl {0, 1}**

Returns : **Current setting**

Description

This command does not perform any function. It is included for legacy compatibility only.

Examples:

```
pl1 <CR>
```

1

```
pl0 <CR>
```

0

5.33   Set Response Modes (rm)

Command Type   : **Legacy/New**
Syntax   : **rm <verbose{0, 1}>[, <new{0, 1}>]**

Returns   : **OK or 0**

Description

      This command is used to set the response modes. The first argument can be 0 or 1, representing terse mode or verbose mode, respectively. In verbose mode all characters sent to the device are echoed back out and most commands return confirmations and full status strings. In terse mode, no characters are echoed and command responses are limited to error codes for legacy behavior. The second argument is optional (if not supplied it defaults to 0), and indicates the protocol mode – legacy (0) or new (1). Legacy mode maintains protocol compatibility with libraries version 15 and earlier.

      The response modes are set on a per-port basis, the change being effective for the port over which the command is sent.

Examples:

```
rm0,1 <CR>
```

OK

```
rm0,0 <CR>
```

0

5.34 Set EEPROM Byte (se)

Command Type   : New
Syntax   : se <byte_number>,<value>

Returns   : DONE

Description

This command will temporarily change an EEPROM setting. The setting is not actually committed to non-volatile memory with this command, and therefore if the device is given a bad setting that causes it to malfunction, removing power will clear the condition. The index of the byte to change is given in <byte_number> and the new value is given by <value>. The index is given as a decimal number and the value is given as a hexadecimal number.

The commonly configured parameters are almost always available through other commands, so this command is used to alter the remaining parameters. These parameters are often low-level timing configurations. Contact Birger Engineering if your programming needs might require modification to these parameters.

Example:

```
se4,0 <CR>
```

DONE

5.35   Set Focus Counter Offset (sf)

Command Type   : Legacy
Syntax   : sf <count>

Returns   : OK

Description

      This command sets the focus counter to mark the current lens focus position with the value specified. For example, moving the focus to the zero position using the 'mz' command will yield the lens' encoder position for the zero stop, which will almost always be non-zero. Using this command with an argument of zero at that point allows you to mark that position as zero and then zero would be the position input to the focus absolute commands to return to that same position.

Example:

sf0 <CR>

OK

5.36 Set GPIO (sg)

Command Type : New
Syntax : sg <port>,<state>

Returns : DONE

Description

This command is used to set the state of the GPIO signals from the device.

Port: 0 : Signal RA0
1 : Signal RA1

State: 0 : Output low
1 : Output high
2 : Input

Examples:

```
sg0,0 <CR>
```

DONE

```
sg1,2 <CR>
```

DONE

5.37   Set Special Modes (sm)

Command Type   : New
Syntax   : sm <mode_flags>

Returns   : DONE

Description

      This command is used to control special modes of the device. <mode_flags> is given as a decimal number. If you need to know more about how the modes work, contact Birger Engineering, Inc.

Bit	*Mode*
0	Reserved.
1	Reserved.
2	1 = Switch 'ma' command to use the servo aperture routine.
3	1 = Turn on background lens querying.
4	1 = Show distance stop progress.
5	1 = Set focus mode for sticky lenses.
6	Reserved.
7	Reserved.

Examples:

```
sm4 <CR>
```

DONE

```
sm12 <CR>
```

DONE

5.38   Print Serial Number (sn)

Command Type   : **Legacy/New**
Syntax   : **sn**

Returns   : **Device serial number**

Description

      This command prints the device serial number. The serial number is used as an input to various commands as an unlock code. This command does not appear in versions 16 and 17 of the library, so obtaining the serial number requires exiting to the bootloader for those versions. Beginning with library version 19r the serial number is printed with leading zeros in a five-digit field.

Example:

```
sn <CR>
```

01234

5.39 Set Spontaneous Responses (sr)

Command Type   : New
Syntax   : sr {0, 1}

Returns   : OK

Description

   This command turns spontaneous responses on or off for the port over which the command is sent; 0 for off, 1 for on. For status updates, focus and aperture changes will be automatically broadcast to every other port in response to a change on any particular port. If background lens querying is also turned on (see command 'sm') then changes in lens status (focal length, etc.) are broadcast to all ports. Finally, GPIO changes will also be broadcast to all ports.
   Note that this does not cause commands to be echoed out, rather updates take the form of status strings as given in Section 3.9. Spontaneous updates are set on a port-by-port basis. The status changes are broadcast, but only the communications ports over which this command has been sent to turn on responses will actually receive them.

Examples:

sr0 <CR>

OK

sr1 <CR>

OK

5.40   Short Version String (vs)

Command Type   : New
Syntax   : vs

Returns   : s:\<module>v\<version>

Description

This command prints the short version string that describes the code currently executing. This string is given with the identifier "s:" followed by the short module ID in \<module>, followed by the letter 'v', followed by the decimal revision number. The Canon EF-232 Library will return 'C2' as the module ID.

Example:

```
vs <CR>

s:C2v17
```

5.41 Write EEPROM Parameters (we)

Command Type   : New
Syntax   : we <serial_number>

Returns   : DONE

Description

       This command will commit the temporary non-volatile settings to EEPROM. This will make the settings permanent. You may damage your device, which will require sending it back to Birger Engineering, if you commit bad parameters to the non-volatile memory. This command requires the device serial number as a confirmation. The serial number is parsed numerically, so leading zeros are not significant.

Example:

```
we1234 <CR>
```

DONE

6   Bootloader Command Reference

This section describes commands available while the bootloader is active. These commands are not available while the library is executing, although the library may have some of the same commands also implemented.

6.1 Banner Echo (be)

Command Type : Legacy
Syntax : be

Returns : Birger Engineering, Inc.

Description

This command prints the device banner string.

Example:

```
be <CR>

Birger Engineering, Inc.
```

6.2 Boot Library (bl)

Command Type : New
Syntax : **bl**

Returns : **Depends on response mode settings**

Description

This command boots the library code. Control is transferred to the library and the library handles all further command processing.

Example:

```
bl <CR>

BEI Boot Executive, v44
(c)2008 Birger Engineering, Inc.
Canon EF-232 Library v17
```

6.3   Baud Rate (br)

Command Type   : Legacy
Syntax   : br <baud_rate_div_100>[,<port>]

Returns   : DONE

Description

      This command sets the baud rate for the current port, unless <port> is specified (the port can be specified only in bootloader versions 45 and above). The baud rate is specified as the desired baud rate divided by 100 in <baud_rate_div_100>. The confirmation is printed in the new baud rate if the current port has been changed.
      The baud rate is set on a per-port basis.

Port: 0   : Secondary serial port
       1   : Primary serial port
       2   : Auxiliary serial port 1
       3   : Auxiliary serial port 0
       4   : USB

Examples:

```
br1152 <CR>

DONE (at baud 115,200)

br96,2 <CR>

DONE (at current baud – but port 2 is set to 9600)
```

6.4   Baud Store (bs)

Command Type   : Legacy
Syntax   : bs

Returns   : DONE

Description

    This command will commit the current baud rates to non-volatile memory. This command was removed from bootloader versions 44 and above. Instead the write configurations command 'wc' is used.

Example:

```
bs <CR>
```

DONE

6.5   Hardware Version (hv)

Command Type   : Legacy
Syntax   : hv

Returns   : Device hardware revision

Description

This command prints the hardware revision of the device.

Example:

```
hv <CR>
```

4

6.6   Library Version (lv)

Command Type   : Legacy
Syntax   : lv

Returns   : **Full library identity string**

Description

This command prints the full library identification string of the library currently in memory. This command can be used after a firmware download to verify that the new library was loaded and programmed correctly.

Example:

```
lv <CR>
```

Canon EF-232 Library v17

6.7   Query Communications Port (qp)

Command Type   : New
Syntax   : **qp**

Returns   : **Port number**

Description

  This command prints the identification number of the port over which communications are occurring.

Port: 0   : Secondary serial port
        1   : Primary serial port
        2   : Auxiliary serial port 1
        3   : Auxiliary serial port 0
        4   : USB

Example:

qp <CR>

4

6.8   Set Response Modes (rm)

Command Type   : Legacy/New
Syntax   : rm <verbose{0, 1}>[, <new{0, 1}>[,<port>]]

Returns   : OK or 0

Description

      This command is used to set the response modes. The first argument can be 0 or 1, representing terse mode or verbose mode, respectively. In verbose mode all characters sent to the device are echoed back out and most commands return confirmations and full status strings. In terse mode, no characters are echoed and command responses are limited to error codes for legacy behavior. The second argument is optional (if not supplied it defaults to 0), and indicates the protocol mode – legacy or new. Legacy mode maintains protocol compatibility with libraries version 15 and earlier.

      The response modes set in the bootloader are copied forward into the library when the library is booted. They can be committed to non-volatile memory in the bootloader so that the desired settings are active when power is applied to the device.

      The response mode is set on a per-port basis, the settings being active for the port over which the command is sent unless the third argument is specified (only in bootloader versions 45 and above). In order to emulate legacy behavior, if the command protocol is set to legacy (0) then an automatic write of this parameter, and consequently all other non-volatile parameters, occurs.

Port: 0   : Secondary serial port
      1   : Primary serial port
      2   : Auxiliary serial port 1
      3   : Auxiliary serial port 0
      4   : USB

Examples:

rm0 <CR>

0

rm0,1 <CR>

OK

rm1,0,2 <CR>

OK

6.9   Escape from Port Routing (routeesc,)

Command Type   : New
Syntax   : routeesc,<depth>

Returns   : Routing canceled

Description

    This command is used to cancel port routing. The depth of the route was previously assigned using the port routing command '<u>rt</u>'. If the depth given in this command is less than or equal to the previously set depth then the routing will be canceled.
    This command must be entered exactly as shown; no white-space is allowed.

Example:

```
routeesc,0<CR>
```

Routing canceled

6.10   Setup Port Routing (rt)

Command Type   : New
Syntax   : rt <dest_port>,<depth>

Returns   : <source_port><-><dest_port>:<depth>

Description

     This command redirects all communications from the current port to the port specified in <dest_port>. <depth> is a number from 0-9 assigned to the connection solely as a reference tag to be used in the routing escape command.

Port: 0   : Secondary serial port
        1   : Primary serial port
        2   : Auxiliary serial port 1
        3   : Auxiliary serial port 0
        4   : USB

Examples:

rt0,0 <CR>

1<->0:0

rt2,4 <CR>

1<->2:4

6.11   Serial Number (sn)

Command Type   : **Legacy**
Syntax   : **sn**

Returns   : **Device serial number**

Description

This command prints the device serial number. The serial number is used as an input to the firmware download command as an unlock code. Beginning with bootloader version 45 the serial number is printed with leading zeros in a five-digit field.

Example:

```
sn <CR>
```

01234

6.12 Suppress Prompt (sp)

Command Type   : New
Syntax   : sp {0, 1}

Returns   : DONE

Description

   This command disables the command prompt. A setting of 0 causes the command prompt to appear, and a setting of 1 turns the prompt off. When the prompt is on, a single greater-than symbol is printed as a prompt whenever the device is ready to accept a command. This setting is copied forward into the library when booted. The factory setting is to have the prompt disabled.
   The command prompt is enabled/disabled on a per-port basis, the setting being active for the port over which the command is sent.

Example:

```
sp0 <CR>
```

OK
>

6.13   Long Version String (vn)

Command Type   : New
Syntax   : vn

Returns   : **Long version string**

Description

   This command prints the long version string for the device.

Example:

```
vn <CR>

BEI Boot Executive, v44
(c)2008 Birger Engineering, Inc.
```

6.14   Short Version String (vs)

Command Type   : Legacy/New
Syntax   : vs

Returns   : s:<module>v<version>

Description

    Legacy: Prior to bootloader version 44, this command prints an extended version string.

    New: This command prints the short version string that describes the code currently executing. This string is given with the identifier "s:" followed by the short module ID in <module>, followed by the letter 'v', followed by the decimal revision number. The bootloader will return 'BE' as the module ID.

Example:

```
vs <CR>
```

s:BEv44

6.15 Write Configurations to EEPROM (wc)

Command Type : New
Syntax : wc

Returns : OK

Description

      This command will commit the temporary non-volatile settings to EEPROM. This will make the settings persist through power cycles. The parameters that will be committed are the baud rates, response modes, and command prompt settings for all ports, not just the port over which this command was sent.

Example:

```
wc <CR>
```

OK

6.16   XModem Firmware Download (xm)

Command Type   : New
Syntax   : xm <serial_number>

Returns   : Enters XModem download

Description

      This command is used to update the library firmware. As a safeguard against inadvertently entering XModem download mode, the device's serial number must be supplied as the command argument. The serial number is parsed numerically, so leading zeros are not significant. The serial number is obtained with the 'sn' command.

      Any terminal program that supports XModem with 128-byte blocks and simple checksumming (not CRC-16) can be used to download new library firmware. Library firmware is obtained solely from Birger Engineering, Inc.

Example:

```
xm1234 <CR>
```