GHI Electronics, LLC
501 E. Whitcomb Ave.
Madison Heights, Michigan 48071
Phone: (248) 397-8856
Fax: (248) 397-8890
www.ghielectronics.com

Preliminary File System User Manual

Where Hardware Meets Software

1 Contents

2 Introduction 3
3 Commands 4
3.1 I – Initialize Media 4
3.2 O – Open File 5
3.3 W – Write File 5
3.4 R – Read File 6
3.5 F – Flush File 6
3.6 C – Close File 6
3.7 P – Seek File 7
3.8 Y – Tell File 7
3.9 D – Delete File 7
3.10 L – Fast Write 7
3.11 ? – Find File 8
3.12 @ – Prepare File Listing 8
3.13 N – Next Result 9
3.14 A – Rename File 9
3.15 M – Copy File 9
3.16 K – Get Free Size 10
3.17 Q – Format Drive 10
3.18 T – Initialize Real Time Clock 10
3.19 S – Set Date and Time 11
3.20 G- Get Date and Time 11
3.21 Z – Sleep 11
3.22 B – Set Baud Rate 12
3.23 V – Get Version Number 12
3.24 # – Enable Echo 12
3.25 E – Test Speed 12
3.26 J – Get Status 13
4 Serial Interfaces 14
4.1 UART Interface 14
4.2 SPI Interface 14
4.3 I2C Interface 15
5 Operating Modes 16
5.1 SD Reader 16
5.2 Keyboards 17
6 Result Codes 18
7 Legal Notice 19
7.1 Trademarks 19
7.2 Disclaimer 19
8 Revision History 20

2 Introduction

GHI Electronics's file-system solutions offer dedicated coprocessors that combine robust file-system and media access drivers with a complete command set to access and manipulate files through a serial interface. The command set is human readable and simple to parse allowing any basic microcontroller to easily access file systems on removable media. Advanced microcontrollers can also benefit by offloading file operations onto the file-system coprocessor.

Some devices may include extended features, such as the ability to read USB drives or keyboards or simulate an SD card reader. This manual covers the standard commands and features. Each individual product's datasheet has further details on advanced or extended uses.

3 Commands

The commands are designed to be human readable ASCII, yet easily parsed by a microcontroller. Commands are sent to the device over one of the supported serial interfaces and are followed by one or more responses. A response must be received before the next command is issued.

Numbers used in commands and responses are hexadecimal without the "0x" prefix. Numbers sent from the device to the host may be padded with leading zeros.

Each command begins with a single character that identifies the command. Any command parameters follow and are separated from the command identifier character by a space. Commands are terminated with a line-feed character.

Once a command is received, a response is sent consisting of an exclamation point followed by a two-digit hex result code terminated by a line-feed. See the result code table for details. Some commands return additional information. See the specific command for details.

In the tables below, the command format is given. Each command begins with the single identifying character. Mandatory parameters are enclosed in curly braces and optional parameters are enclosed in square brackets, though the brackets and braces are not actually sent to the device. If an optional parameter is preceded by a ">" and the parameter will not be sent, the ">" must also not be sent. When a parameter does not list any valid values, the parameter letter itself is sent. Any other characters present in the command format must be sent as is. Responses are highlighted in red. The parameter "R" is always the result code from the device.

3.1 I – Initialize Media

Command	**I {X}:[H][K]>[D]** **!{R}**	Initializes the specified media. Must be done before interacting with the media.
Parameters	**X**	The media to initialize. • M: SD/MMC memory card • U0: USB0 full-speed • U1: USB1 full-speed • D: USB1 SD-reader (USB client) • K0: USB0 as keyboard reader • K1: USB1 as keyboard reader
	H	For media U1 and D only, use high-speed mode. Must be set with U1 when a ULPI PHY is used regardless of the USB client.
	K	For media K0 or K1 only, return an ASCII-mapped key-code.
	D	For media M, use the specified clock frequency. Each device may have a different default. • 0: 24 MHz • 1: 16 MHz • 2: 12 MHz • 3: 9.6 MHz • 4: 8 MHz

Examples	**I M:** **!00**	Initialize the memory card.
	I D:H **!00**	Initialize USB in SD-reader mode using high-speed mode.
	I M:>1 **!00**	Initialize the memory card with a clock frequency of 16 MHz.

3.2 O – Open File

Command	**O {H}{M}>{N}** **!{R}**	Opens the file in the given mode on the handle. Drive K0 and K1 are automatically opened when initialized and assigned handles Z and Y respectively.
Parameters	**H**	The file handle to use. Must be between 0 and F.
	M	The mode in which to open the file. • R: open for read. Must already exist. • W: open for write. Will be overwritten if exists, create new otherwise. • A: open for append. Starts at the end of the existing file. Creates the file if it does not already exist.
	N	The file to open. The full path must be specified.
Examples	**O 1R>U1:\Test.txt** **!00**	Opens Test.txt on drive U1 for read on handle 1.
	O 0W>M:\A\Test.txt **!00**	Opens Test.txt on drive M in folder A for write on handle 0.

3.3 W – Write File

Command	**W {H}>{N}** **!{R}** **{D}** **${C}** **!{R}**	Writes data to the given handle. After sending the command, wait for the response before sending the actual data.
Parameters	**H**	The file handle to write to.
	N	The number of bytes to write.
	D	The data to write. Must be {N} bytes long.
	C	The actual number of bytes written.
Examples	**W 1>10** **!00** **0123456789ABCDEF** **$00000010** **!00**	Writes 16 bytes to file handle 1.

3.4 R – Read File

Command	**R {H}{F}>{N}** **!{R}** **{D}** **${C}** **!{R}**	Reads data from the given handle. If there are less than {N} bytes left in the file, {F} is sent back after the actual data to pad the result to the required length. Subsequent calls to read will start where the last call left off. When reading from a keyboard, each returned byte represents the key code or ASCII value (depending on the initialization setting) of a pressed key.
Parameters	**H**	The file handle to read from.
	F	The filler byte that is used to pad the data.
	N	The number of bytes to read.
	D	The returned data.
	C	The actual number of bytes read.
Examples	**R 1Z>5** **!00** **01234** **$00000005** **!00**	Reads 5 bytes from file handle 1 with Z as the filler byte.
	R 0Z>5 **!00** **01ZZZ** **$00000002** **!00**	Reads 5 bytes from file handle 0 with Z as the filler byte. The file only has 2 bytes, so the final 3 bytes are Z.

3.5 F – Flush File

Command	**F {H}** **!{R}**	Flushes the data in the internal buffers to the storage device. Received data are buffered to prolong device's life span and increase write speed. Actual writing to the media happens only when necessary or when the Flush command is used.
Parameters	**H**	The file handle to flush.
Examples	**F 3** **!00**	Flush file handle 3.

3.6 C – Close File

Command	**C {H}** **!{R}**	Flushes the internal buffers for the given handle and then closes the file handle so it is available for use again.
Parameters	**H**	The file handle to close.
Examples	**C 4** **!00**	Close file handle 4.

3.7 P – Seek File

Command	**P {H}>{I}** **!{R}**	Sets the position in the file that the next read will start from.
Parameters	**H**	The file handle to seek.
	I	The new position. For files opened in write mode, the only valid position is 0.
Examples	**P 0>10** **!00**	Sets the read position to 0x10 (byte 16) in file handle 0.

3.8 Y – Tell File

Command	**Y {H}** **!{R}** **${I}** **!{R}**	Queries the current read position for the given handle.
Parameters	**H**	The file handle to query.
	I	The position that the next read will start from.
Examples	**Y 1** **!00** **$00000005** **!00**	Requests the read position from file handle 1.

3.9 D – Delete File

Command	**D {F}** **!{R}**	Deletes the given file or folder. The full path must be specified. It must not be in use. Folders must be empty before deleting.
Parameters	**F**	The file or folder to delete.
Examples	**D U1:\Test.txt** **!00**	Deletes Test.txt on drive U1.
	D M:\A\Test.txt **!00**	Deletes Test.txt on drive M in folder A.

3.10 L – Fast Write

Command	**L {H}>{N}** **!{R}** **{D}** **${C}** **!{R}**	Writes data to the given handle. After sending the command, make sure to read the initial response before sending the data. This command is only available over the SPI interface and is not valid with keyboards. It is identical to the normal write command except that it achieves faster performance and the data itself is not encapsulated in a write frame – it is sent as-is. When sending the data, send it in exactly 8,192 byte chunks. Only the last chunk may be smaller than 8,192 bytes. Check the BUSY pin between sends.

Parameters	**H**	The file handle to write to.
	N	The number of bytes to write.
	D	The data to write. Must be {N} bytes long.
	C	The actual number of bytes written.
Examples	**L 1>10** **!00** **0123456789ABCDEF** **$00000010** **!00**	Writes 16 bytes to file handle 1.

3.11 ? – Find File

Command	**? {N}** **!{R}** **${S}** **${A}** **${M}** **!{R}**	Queries for information about the specified file or folder, if it exists.
Parameters	**N**	The file or folder name, including the full path, to query.
	S	The size of the file in bytes.
	A	The attributes of the file represented as 8 bit flags. • Bit 0: read only • Bit 1: hidden • Bit 2: system • Bit 3: volume ID • Bit 4: folder • Bit 5: archive • Bit 6: reserved • Bit 7: reserved
	M	The last modified date and time. Given in the format {hh:mm:ss mm-dd-yyyy}.
Examples	**? U1:\Test.txt** **!00** **$00000F34** **$00** **$13:00:00 01-24-2011** **!00**	Queries information about Test.txt on drive U1. It is 3,892 bytes in size. It is not marked as read only, hidden, system, volume ID, folder, or archived. It was last modified at 13:00:00 on 01-24-2011.

3.12 @ – Prepare File Listing

Command	**@ {F}** **!{R}**	Prepares to list all of the files and folders in the given folder. The full path is required. Results are returned using the N – Next Result command.

Parameters	**F**	The folder whose contents will be listed.
Examples	**@ M:\Test\Foo** **!00**	Prepares to list all of the files and folders in the Foo folder.

3.13 N – Next Result

Command	**N** **!{R}** **${N}** **${A}** **${S}** **!{R}**	Returns the next result from the @ – Prepare File Listing command. Only one result is returned at a time, so this command must be repeated to get all of the results.
Parameters	**N**	The file or folder name.
	A	The attributes of the file represented as 8 bit flags. See the attributes under the ? – Find File command.
	S	The size of the file in bytes.
Examples	**N** **!00** **Test.txt** **$00** **$00000F34** **!00**	Returns the next file. It is called Test.txt, is 3,892 bytes in size, and has no attributes.

3.14 A – Rename File

Command	**A {O}>{N}** **!{R}**	Renames the given file.
Parameters	**O**	The file to rename. The full path must be included.
	N	The new file name. Do not include the path.
Examples	**A U1:\A.txt>B.txt** **!00**	Renames A.txt on U1 to B.txt.

3.15 M – Copy File

Command	**M {S} {I} {D} {C}** **!{R}** **${A}** **!{R}**	Copies data from one handle to the next.
Parameters	**S**	The file handle to copy from.
	I	The offset in the source file to begin copying from.
	D	The file handle to copy to.
	C	The number of bytes to copy from the source.

	A	The number of bytes actually copied.
Examples	**M 0 0 1 64** **!00** **$00000064** **!00**	Copies 0x64 bytes starting at position 0 from file handle 0 to file handle 1.

3.16 K – Get Free Size

Command	**K {D}** **!00** **${S}** **!00**	Queries how many bytes are available on the specified drive. It may take several seconds to calculate.
Parameters	**D**	The drive to query. See the I – Initialize Media command for available drives.
	S	The number of bytes free on regular drives and the number of bytes available to read for keyboards.
Examples	**K U1:** **!00** **$0000000000000064** **!00**	Queries how many bytes are available on U1. There are 100 bytes available.

3.17 Q – Format Drive

Command	**Q CONFIRM FORMAT {D}** **!{R}** **!{R}**	Formats the given drive. CONFIRM FORMAT is required. The first response is returned immediately and the second once formatting is complete.
Parameters	**D**	The drive to query. See the I – Initialize Media command for available drives.
Examples	**Q CONFIRM FORMAT M:** **!00** **!00**	Formats drive M:.

3.18 T – Initialize Real Time Clock

Command	**T {M}** **!{R}**	Initializes the Real Time Clock (RTC) to use the given mode.
Parameters	**M**	The mode to use. • S: shared mode. Runs off of the processor clock. • B: backup mode. Runs off of an external crystal and VBAT.
Examples	**T S** **!00**	Initializes the RTC in shared mode.

3.19 S – Set Date and Time

Command	**S {T}** **!{R}**	Sets the system date and time.
Parameters	**T**	The new date and time in FAT file system format. • Bits 0 – 4: Second divided by 2 (0 – 30) • Bits 5 – 10: Minute (0 – 59) • Bits 11 – 15: Hour (0 – 23) • Bits 16 – 20: Day (1 – 31) • Bits 21 – 24: Month (1 – 12) • Bits 25 – 31: Years since 1980.
Examples	**S 34212002** **!00**	Sets the system date and time to 04:00:04 1/1/2006.

3.20 G- Get Date and Time

Command	**G {C}** **{T}** **!{R}**	Gets the system date or time.
Parameters	**C**	The component to return. • D: return the system date. • T: return the system time.
	T	The system date in the format {mm-dd-yyyy} or system time in the format {hh:mm:ss}.
Examples	**G D** **01-01-2006** **!00**	Gets the system date which is 01/01/2006.

3.21 Z – Sleep

Command	**Z {C}>[S]** **!{R}**	Executes various system control functions.
Parameters	**C**	The function to execute. • 0: enter standby mode. Woken by a rising edge on the wakeup pin or a reset. The system is reset when exiting this mode. • 1: enter stop mode. Woken by a rising edge on the wakeup pin. The system resumes execution when exiting this mode. • 2: Changes the function of the wakeup pin. • 3: Reboot in the low-level bootloader.
	S	Valid for function 2 only. • 0: Wakeup pin functions as wakeup. • 1: Wakeup pin functions as an emergency flush and close on a rising edge. The system must be reset after use.

Examples	**Z 3** **!00**	Reboots into the low-level bootloader.

3.22 B – Set Baud Rate

Command	**B {N}** **!{R}** **!{R}**	Sets the baud rate for the UART interface. The first response is sent at the current baud rate and the second response is sent at the second baud rate. This setting does not survive a system reset.
Parameters	**N**	The new baud rate.
Examples	**B 1C200** **!00** **!00**	Sets the baud rate to 115,200.

3.23 V – Get Version Number

Command	**V** **{V}** **!{R}**	Queries the firmware version number.
Parameters	**V**	The firmware version.
Examples	**V** **V2.0.0** **!00**	Queries the system version which is v2.0.0.

3.24 # – Enable Echo

Command	**# {E}** **!{R}**	Controls the echo of everything sent to the system. Echo is disabled by default.
Parameters	**E**	The mode to set. • 0: disable echo. • 1: enable echo.
Examples	**# 1** **!00**	Enables echo.

3.25 E – Test Speed

Command	**E {D}>{S}** **!{R}** **${A}** **${B}** **!{R}**	Tests the read and write speed of the specified drive. The drive must be initialized. Large test sizes can take several minutes to complete.
Parameters	**D**	The drive to test. See the I – Initialize Media command for available drives.
	S	The number of bytes to test. Must be a multiple of 1,024.

	A	How long it took to write the specified number of bytes in milliseconds.
	B	How long it took to read the specified number of bytes in milliseconds.
Examples	**E M:>6400000** **!00** **$00005D14** **$000039FB** **!00**	Tests drive M: by read and writing 100MB. It took 23,828 ms to write and 14,843 ms to read.

3.26 J – Get Status

Command	**J S** **!{R}** **${S}** **!{R}**	Queries various status registers.
Parameters	**S**	The register to query. • 0: device status. ○ Bit 0: 1 = Card Detect (CD) is set, 0 = CD is not set ○ Bit 1: 1 = Write Protect (WP) is set, 0 = WP is not set ○ Bit 2: 1 = U0: is mounted, 0 = U0: is not mounted ○ Bit 3: 1 = U1: is mounted, 0 = U1: is not mounted ○ Bit 4: 1 = U1: is high-speed, 0 = U1: is full-speed ○ Bit 5: 1 = U0: is present, 0 = U0: is not present ○ Bit 6: 1 = U1: is present, 0 = U1: is not present ○ Bit 7: 1 = U1: is in SD reader mode, 0 = U1: is in host mode • 1: file status. Bits 0 through 15 represent whether or not file handles 0 through 15 respectively are opened or not. 1 is opened, 0 is closed. • 2: auxiliary status. ○ Bit 0: 1 = U0: is in keyboard mode, 0 = U0: is in mass storage mode. ○ Bit 1: 1 = U1: is in keyboard mode, 0 = U1: is in mass storage mode. ○ Bit 2: 1 = K0: is returning raw values, 0 = K0: is returning filtered ASCII ○ Bit 3: 1 = K1: is returning raw values, 0 = K1: is returning filtered ASCII ○ Bit 4 – 15: reserved
Examples	**J 1** **!00** **$0101** **!00**	Queries register 1. File handles 0 and 8 are in use.

4 Serial Interfaces

Available communications options are UART, SPI, and I2C. All commands, responses, and operations are identical no matter which interface is used. Each interface only differs in how the data is transferred.

The interface is selected on power up based on the state of specific device dependent pins. See the device specifications for details.

For large write payloads, the host must verify the device is ready to receive more data by checking the BUSY pin. When the device is busy, the host must wait and try again.

4.1 UART Interface

The UART interface uses three signals:

- RX: serial input that receives commands from the host.
- TX: serial output, the responses going to the host microcontroller.
- BUSY: output indicating when the system is busy and cannot accept more incoming data.

4.2 SPI Interface

The SPI interface uses six signals:

- SSEL: Slave select input.
- SCK: Serial clock input.
- MISO: Master in slave out, output.
- MOSI: Master out slave in, input.
- BUSY: output indicating when the system is busy and cannot accept more incoming data.
- DATA READY: output indicating there is data available to read.

There is a framing mechanism used to transfer data to and from the host. Each frame can include part of or multiple commands and responses. Every frame originates from the host and includes a three byte header. The first byte determines if this is a read or write frame (1 for write, 2 for read) and the last two bytes are the payload size (excluding the header), least-significant byte first.

Instead of using the BUSY pin, the host can check the response sent on MISO. For every byte sent on MOSI after the frame type byte, the device sends a ready flag byte on MISO. If the byte is not 1, the device is busy.

Keep in mind that SPI is a host initiated protocol. The host must query the device using a read frame to see if it has data it needs to send. The payload size field in the read frame is the number of bytes the host wants to read from the device. At the same time that the payload size field is sent, the device will send how many bytes it has available in its internal buffer on MISO. If the device has fewer bytes available than the host requested, make sure to only read the number of bytes the device actually has available.

Pin	Frame Type	Payload Size		Payload		
MOSI	0x01 (Write)	Size LSB	Size MSB	Byte 0	…	Byte N
MISO	0x00	Ready Flag	Ready Flag	Ready Flag	Ready Flag	Ready Flag

Pin	Frame Type	Request Size		Request		
MOSI	0x02 (Read)	Requested Bytes LSB	Requested Bytes MSB	0x00	0x00	0x00
MISO	0x00	Available Bytes LSB	Available Bytes MSB	Byte 0	…	Byte N

4.3 I2C Interface

The I2C interface uses four signals:

- SCL: I2C clock.
- SDA: I2C data.
- BUSY: output indicating when the system is busy and cannot accept more incoming data.
- DATA READY: output indicating there is data available to read.

Reading and write data over I2C uses the built in I2C read and write commands. When reading, bytes 0x00 and 0xFF have special meaning. All other bytes are to be interpreted as part of the response. When 0x00 is received, no data is available.

0xFF is an escape byte. When it is received, it means the next byte is to be interpreted as-is instead of any special meaning. This includes 0x00 and 0xFF. For example: when 0xFF is received followed by 0x00, 0x00 was received as part of the actual response. 0xFF followed by 0xFF means 0xFF was received as part of the actual response. For commands that have a filler byte being sent to the host, do not use 0xFF or 0x00 as the filler byte.

The following code illustrates reading and handling the escape bytes:

```
uint8 I2C_ReadByteFromDevice() {
        uint8 data;

        do {
                I2C_SendAddress((deviceAddress << 1) | 0x01);
                data = I2C_Read();
        while (data == 0x00);

        if (data == 0xFF) {
                I2C_SendAddress((deviceAddress << 1) | 0x01);
                data = I2C_Read();
        }

        return data;
}
```

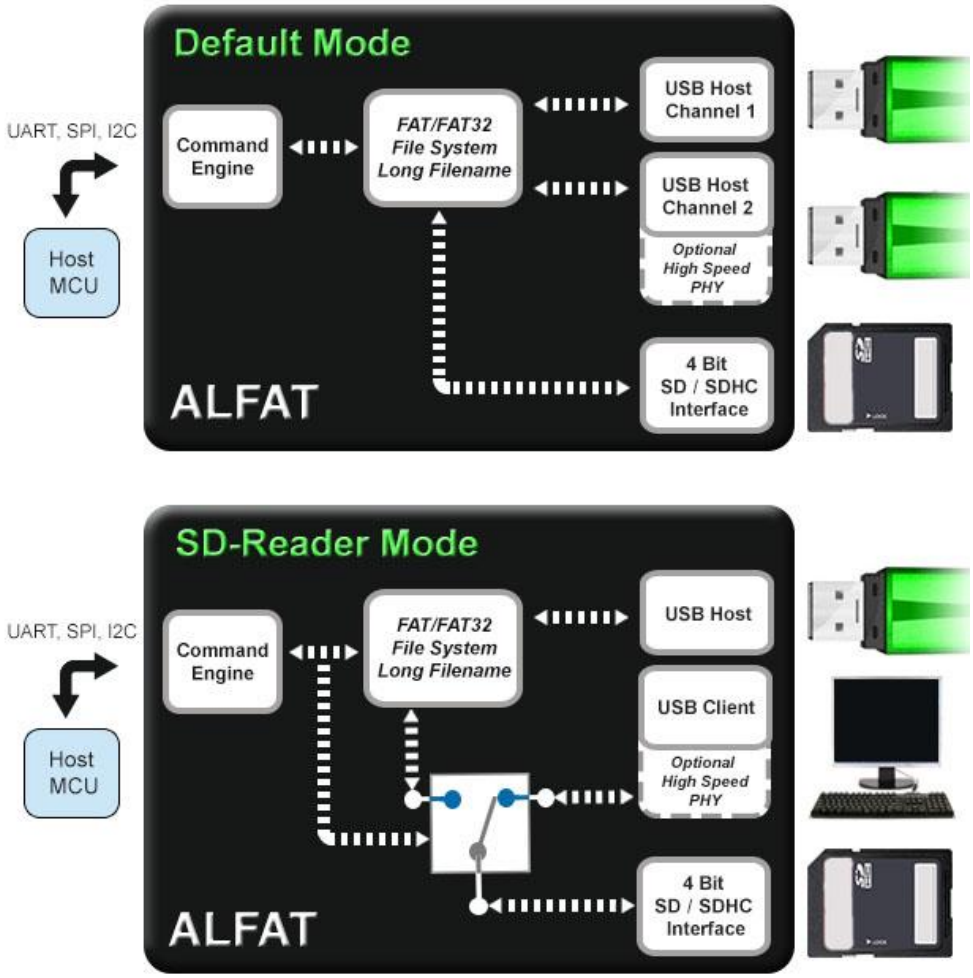
5 Operating Modes

5.1 SD Reader

Some devices include a special SD-Reader mode in addition to the default mode. In the default mode, the system is a file system hardware device, allowing access of files on SD and USB drives over UART, SPI, and I2C.

In the SD-Reader mode, one of the USB ports operates in client mode appearing as a USB SD card reader. In this mode, a product can, for example, store files on an SD card and then use the USB port to read the files from a PC.

Once initialized in SD Reader mode and a valid connection with the PC is established, the SDR Connect pin will go high. It will go low when the PC disconnects or SD Reader mode is exited. Connection notification is not instant and may take some time depending on the connected devices.



5.2 Keyboards

Some devices include the ability to read key presses from an attached USB keyboard. See the R – Read File and K – Get Free Size commands for details on use.

6 Result Codes

Code	Description
0	Command successful.
1	Unknown command.
2	Incorrect parameters.
3	Operation failed or attempted to write to a write-protected card.
4	Reached the end of the file/folder list.
10	Media does not initialize.
11	Initialize media failed. May be caused by not present media or high or floating card detect.
12	Insufficient free space on storage media.
20	File/folder does not exist.
21	Failed to open the file.
22	Seeking a file must be open for read.
23	Seek value outside the file size.
24	Filename length cannot be zero.
25	Filename has a forbidden character.
26	File or folder name already exists.
30	Invalid handle.
31	Handle source cannot be opened.
32	Handle destination cannot be opened.
33	Handle source requires file open for read mode.
34	Handle destination requires file open for write or append mode.
35	No more handles available.
36	Handle cannot be opened.
37	Handle is already in use.
38	Invalid file open mode.
39	Handle requires write or append mode.
3A	Handle requires read mode.
40	The system is busy.
41	Command is supported with SPI interface only.
60	Flush signal was detected.
70	No SD Card detected in card reader mode.
71	Keyboard not detected.
72	Keyboard not initialized.

7 Legal Notice

7.1 Trademarks

F20, F40, and ALFAT are trademarks of GHI Electronics, LLC.

Other registered or unregistered trademarks are owned by their respective companies.

7.2 Disclaimer

IN NO EVENT SHALL GHI ELECTRONICS, LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS PRODUCT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. GHI ELECTRONICS, LLC LINE OF PRODUCTS ARE NOT DESIGNED FOR LIFE SUPPORT APPLICATIONS. SPECIFICATIONS, AVAILABILITY, AND PRICE ARE SUBJECT TO CHANGE WITHOUT NOTICE.

8 Revision History

Revision	Date	Change
3.1	2016-09-23	Corrected ? parameter name.
3.0	2016-06-16	Initial preview release.