

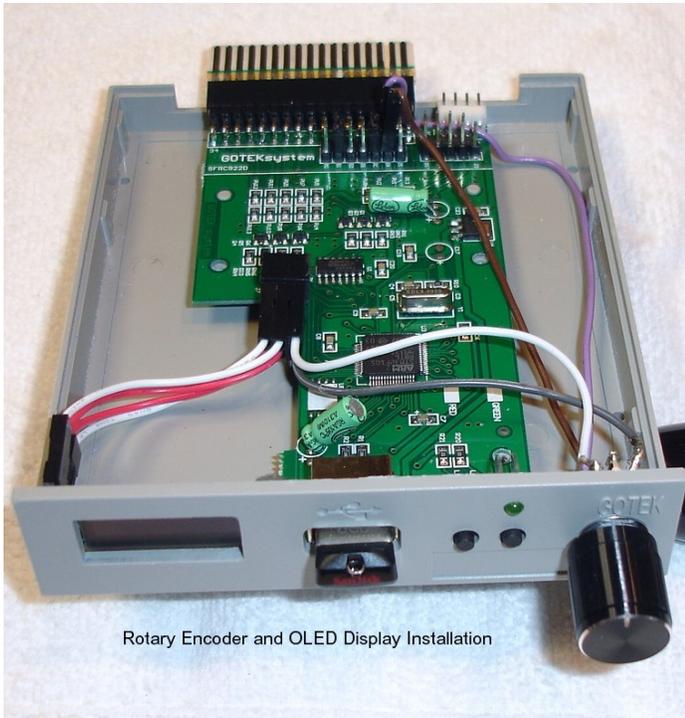


Gotek Floppy Emulator Hardware Upgrades

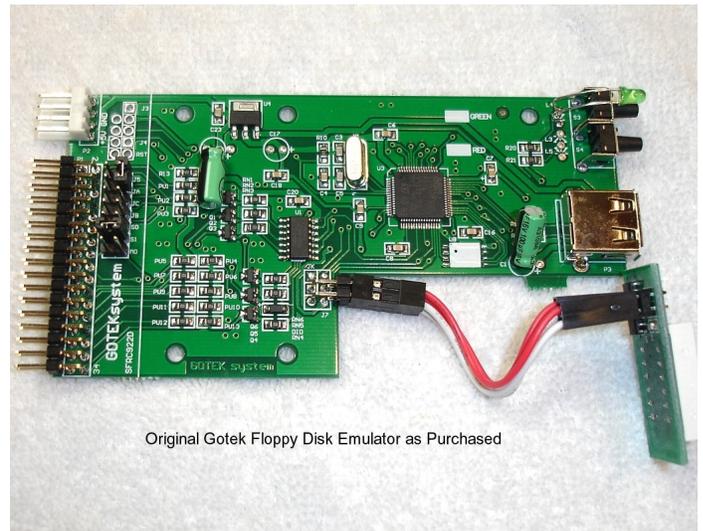
FlashFloppy Gotek Course 301

This article will introduce you to the procedures necessary to upgrade the hardware of the basic Gotek models listed earlier.

Installing the OLED Upgrade



Rotary Encoder and OLED Display Installation



Original Gotek Floppy Disk Emulator as Purchased

Installing the OLED display is fairly simple. You unplug the digital display of the original board (see picture above), then plug the new display into the same connector pins, taking note of the pin labels of the next picture.

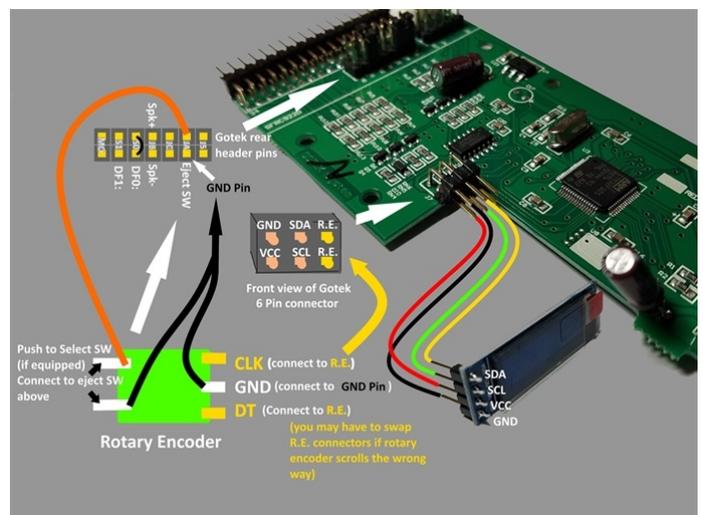
GOTEK Floppy Emulator

In the previous issue we discussed the capabilities, cabling requirements, and limitations of the basic Gotek Floppy Disk Emulator, as bought, including its use in the Z-100 series computer. The models marketed at the time included:

- SFR1M44-U100K code: U00
- SFR1M44-U100K-R code: U01
- SFR1M44-TU100K code: U02

Then, in this issue, we discussed the Firmware Upgrade separately, again including all the ways that we can use the new capabilities in the Z-100. All very impressive.

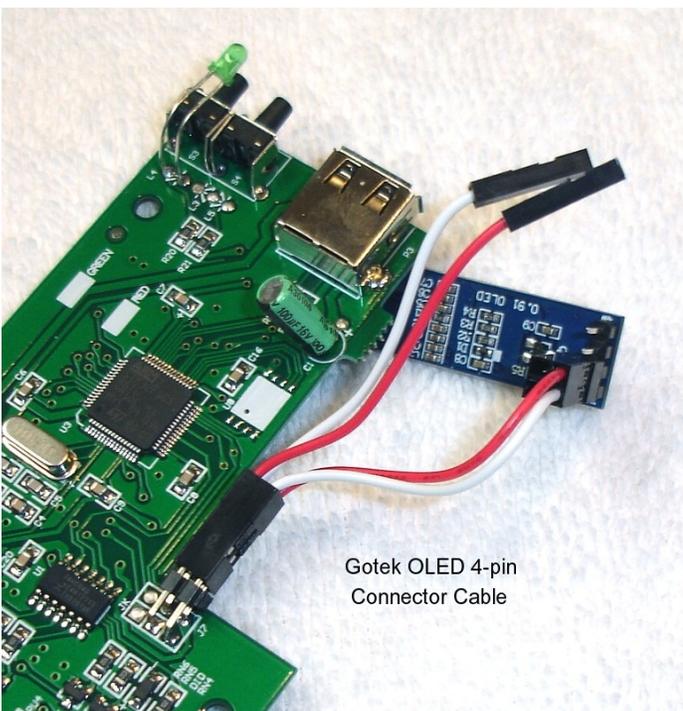
But in this article, we begin to put it all together. And I think you will be as impressed as I am with the final product. You will need to obtain a 0.91" 4P IIC OLED display. There is another that will work, but it is larger in size, and I'm not sure it will fit in the case. Don't worry about the Rotary Encoder for now (read this article first).





Here is how the OLED looked just before installing in the Gotek.

Note: The most recent Gotek that I ordered came with a pair of four-pin connectors instead of two pairs of wires with two-pin connectors. The four-pin connector must be taken apart at one end (there is a plastic tab that holds the metal pins inside the plastic piece; just pry the tab up gently and pull the wire free) and two wires that can be reinserted, but reversed, and the ends of the separate, removed wires placed in single connectors, as shown in the next picture. OR separate jumper wires can be purchased (these come in sets of 40 from E-bay). Please look carefully at the following picture and the above drawing for wire positioning.



I left the four-pin connector with the two open wire bays attached to the OLED as shown, where they are out of the way.

Really anxious to see how the display upgrade would look, I was quick to install the OLED upgrade. I also modified the case (should not have yet) for all the upgrades to be made, but after installing just the OLED display, I wanted to see how it worked.

First, out of curiosity, I tried the OLED in my new, second Gotek that still had the original firmware. As expected, the OLED did nothing. So, the old firmware has no capabilities for handling the OLED or probably even a rotary encoder.

I reinstalled the OLED back on the modified Gotek circuit board.

Note: You will find numerous descriptions of the specific modifications needed to be made physically to the Gotek case on the web, including several YouTube videos, so I will not try to list each procedure here.

For as long as it may be available, my favorite YouTube video is:
<https://www.youtube.com/watch?v=J3OZo261EpA>

but there are many others. Just search for Gotek upgrades. You will need to expand the window for the OLED display in the front panel to the right and remove a tab on the upper cover that supported the old 2 or 3 digit display.

Using the 8GB USB flash drive that I had used to back up files on the **unmodified** Gotek, all the OLED displayed was:

```
[DOS]
000/000 D:0
```

briefly; then changed to:

```
[..]
000/000 D:1
```

The buttons did not work (no surprise) since there is no FF.CFG file to define button use, but I had hoped they would still do something – NO.

I powered down and inserted the 4GB flash drive prepared on the PC for a 720K 3" drive into the **upgraded** Gotek. At power up, the OLED flashed the Firmware version:

```
FlashFloppy
v3.23
```

then the image file name:

```
ZFD720_0000
017/030 HFE D:0
(The D:0 was brief, then becoming T:00.0)
```

The buttons incremented as they should; right to increment ones, left to increment tens. OK

The files incremented to ZFD720_0001, then _0002, then _0024!?! Not sure why at the time, but through more experimentation it appears that only partitions with file names will be displayed and the partition names are rearranged. Empty partitions are skipped.

Then numbers under the Image file name, also incremented; starting at 017/030, to 018/030, to 019/030, then 010/030, 011/030, etc, to repeat up to 019/030.

When it counted up to ZFD720_0030, it jumped back to _0000, because I only used 31 image files.

Ok, I can deal with this, but if I change the <VOL> name, will the OLED show that? NO.

Here's how that went...

After booting to ZDOS v4 the OLED was blank. Because this USB flash drive was prepared as a 720Kb drive, I had to change DRIVECFG to use a LowDen 3" 720K drive as B:. I rebooted again, and this time the OLED remained lit.

I ran FORMAT B:/s/v on partition 0000. That worked fine and the OLED display showed a flashing 'W' and changing Track and Head numbers (The T:00.0 in the display). I used 'DOS4_BOOT' as the volume name, but no change to the OLED.

I copied the rest of the hard drive's Root directory to B: and created a \DOS partition on B:, then copied all the \DOS files to B:\DOS. All worked fine.

I booted to the Gotek (0000) and everything was great, EXCEPT the OLED display continued to only display the image file name.

The ONLY way I have found to change the image names is to place the USB flash drive back in the USB port of the host computer (my PC) and change the image file names in the USB's Root directory!

I found this to be a MAJOR disappointment! But, I continued the hardware upgrade with the rotary encoder, since I have already made the case modifications, and would revisit this unexpected setback with the OLED at another time.

Installing a Sound Speaker

If you have a desire to hear seek noise and other sounds like a real floppy drive would provide, these are also easily made by adding a piezo speaker across pins marked 'Spkr' in the rear header of the Gotek in the first picture. In the Gotek itself the pins are labeled 'JB'.

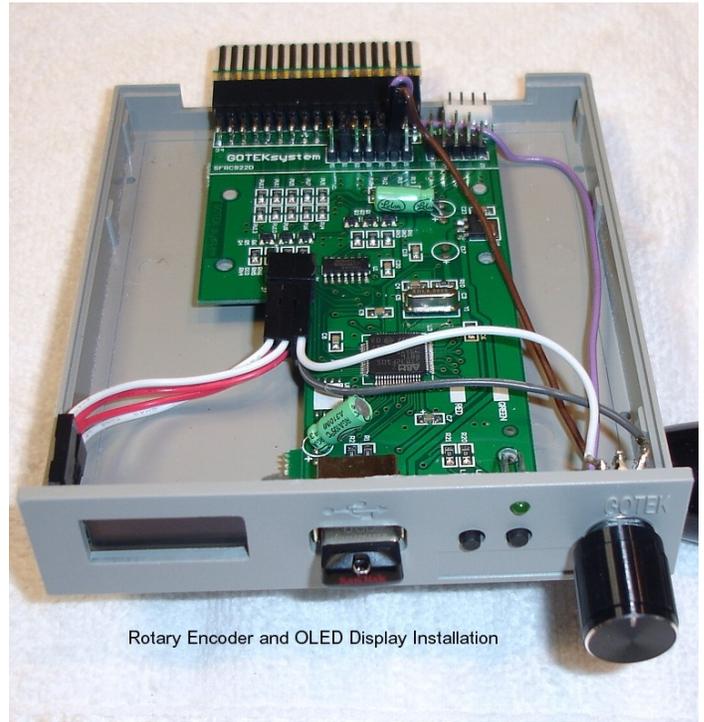
As I didn't feel I needed the floppy disk sounds, I'll leave that upgrade to the user. Again, there are YouTube videos that do add this feature, if you really want it.

Installing a Rotary Encoder

The Rotary Encoders that I ordered did not have the circuit board attached. Those with a circuit board may not fit very well in the Gotek without modification and the only thing that the encoder circuit board provided (some models) was pull-up resistors on the two sides of the rotary part. The rotary encoders are available from E-bay and Amazon. Mine came in a pack of 5.

As with the OLED install, I will not describe the specific installation modifications here. However, please read the following **BEFORE ORDERING** any rotary encoder. I really question its value.

Anyway, an earlier picture above provides the pinout of the rotary encoder. The following pictures show how it appeared in my Gotek.



Rotary Encoder and OLED Display Installation



Rotary Encoder and OLED Display Installation

The second picture also shows the OLED hot glued to the front panel.

The installation went smoothly enough, but when I tested the rotary encoder, it would not change the partition numbers reliably. I checked the rotary settings in the FF.CFG file.

Portion of the FF.CFG file relating to the rotary encoder:

```
# Actions of two navigation buttons.
#           B1      | B2      | Both
# zero:     Prev   | Next   | Slot 0
# eject:     Prev   | Next   | Eject/Insert
# htu:       +10   | +1     | +100
# rotary:    Up-dir | Select/Eject/Insert | -
# rotary-fast: Prev | Next   | Up-dir
# [Prev/Next are accelerated]
# reverse:   Reverse sense of B1 and B2
# Multiple values can be separated by commas,
# eg. twobutton-action=eject,reverse
twobutton-action = htu

# Input sensor type at the rotary-encoder inputs
# (pins PC10 and PC11):
# [full | half | quarter]:
#   Rotary encoder, identified by fraction of
#   a Gray-code cycle performed per detent/click.
#   If default value ('full') requires multiple
#   clicks/detents to move position then change
#   to 'half' (if 2 clicks per move)
#   or 'quarter' (if 4 clicks).
# [trackball]:
#   Blackberry-style trackball
#   (eg. using Hall-effect sensors).
# [buttons]:
#   Push-to-ground Prev/Next buttons.
# [reverse]:
#   If the input is working in reverse,
#   use this option to swap directions.
# [v2]:
#   Use the rotary encoder logic from FlashFloppy
#   v2.x. Use this if the v3 logic is too strict
#   and results in no, or missing, movements.
# Multiple values can be separated by commas,
# eg. rotary=quarter,reverse
# Values: none | quarter | half | full | trackball |
# buttons | reverse | v2
rotary = full
```

In the first section I tried 'htu,rotary', but this messed up the buttons as well as the rotary encoder, so I went back to the default 'htu'.

In the second section, 'Full' is the default position, but I tried the other positions, as well. The best I could get was skipping as many as 11 partition numbers with 'rotary = half'. At 'rotary = full', it rarely worked at all. And at the other settings, the partition numbers just scrolled constantly.

As I had seen that those rotary encoders with a circuit board included pull-up resistors, I tried adding 1Kohm pullup resistors to both sides of the rotary pins, first to 3.3Vdc, then to 5Vdc and while 'half' seemed just a bit more stable at 5Vdc, it did not help in 'full' mode at all. 5Vdc was just barely better than 3.3Vdc.

Thinking it may be feedback between the two resistors via the 5Vdc high side, I even tried increasing the resistor values to 1.5Kohm each, but there was no further improvement. The other modes, 'quarter' and 'v2' just continued scrolling.

I just removed the RE altogether, as it was interfering with access to the buttons. The only thing I lose is the ability to use the eject menu and setting write protection. But I'm not sure that it is worth the effort. If any of you have a positive experience, please let me know, as right now I can NOT recommend even trying it. Perhaps a later firm-ware version will fix this issue. I also do not know if the HxC Firmware has the same problem. In any case, I have not missed it.

To Eject, or Not to Eject

The YouTube videos and other sources make a big deal about the eject and/or disk change (regarding mounting a disk - I have not heard that term, mounting, since 8" drives). In the Z-100, we do not use Disk Change, but rather a Ready Signal. Further, our disk change is detected by other means (I believe that it just times out, but I'm not positive), so by the time you are able to change a disk, the time out forces the computer to assume the disk has changed. That means that the eject function of the rotary encoder, or installation of another separate switch to force an eject of the presently mounted disk, is unnecessary.

I can routinely change the partition numbers in the Gotek by the buttons and the new disk is ready to go. However, there IS ONE EXCEPTION: CP/M requires a {CTRL}-{C} to change and recognize new disks.

Putting it All Together

With the OLED installed and working and the troublesome Rotary Encoder removed, my Gotek was working, but not what I would consider making it worthwhile...

As you recall, before the hardware modifications, I left us with the OLED just showing file names. But, worse than that, the Gotek insisted on rearranging the order of the files...

I changed my blank image files to file names – and the stupid Gotek moved them all alphabetically! Actually, that was done in the PC, even before I copied them to the USB flash drive! And, if you add a file name later, all new file names are placed at the end of the existing file names. So, not only will I lose control of what's where, but I MUST remember the file name of what I want, or scroll through them until I find the right one.

Try to keep track of that with a piece of paper!

Disgusted, I took about two weeks off - the spring chores were calling for action.

Then one morning I woke up at about 0400 (4:00 a.m. for you civilians, but it's still too early, however you consider it) recently thinking about my Gotek and my discouraging results. I had read something quite some time ago about navigation modes and I recalled that one was the Indexed Navigation Mode. I had been using the Native Navigation Mode (supposedly the most popular setting). So I got up to do some additional research.

There are actually three navigation modes available for the Gotek, defined as:

- **Native Mode:** As the default setting, there is no need for any configuration files. It allows direct selection of any valid image file on your USB flash drive.
- **Indexed Mode:** Switches between image names of the form DSKA0000, DSKA0001, etc. This mode is configured within FF.CFG by the command: **nav-mode = indexed**. Each image is automatically assigned to the corresponding numbered partition on the USB Flash Drive in the Gotek. The DSKA prefix can be changed within FF.CFG with the command: **'indexed-prefix= indexed'**

- **HxC Compatibility Mode:** This mode selects between pre-assigned partitions in an HXCSDFE.CFG file. These slots are configured by an image-selector; this is immediately accessible in partition 000 by a file selector. This mode will not be used.

The navigation mode is configured within FF.CFG.

```
# Navigation mode for selecting images/slots
# native: Navigate thru all valid images/directories
# indexed: Navigate through DSKA0000, DSKA0001, ...
# default: native unless overridden by
# HxC compatibility mode configuration
nav-mode = default
```

And can be changed by making:

```
nav-mode = indexed
```

Now, each image is automatically assigned to the corresponding numbered partition on the USB Flash Drive in the Gotek. But it required specific file names of the form "DSKA0000.hfe".

But while the four digit file numbers were required, the "DSKA" prefix could be changed, again within FF.CFG:

```
# Prefix for image names in indexed navigation
# mode. String can be empty ("").
indexed-prefix = "DSKA"
```

And could be modified with the command:

```
indexed-prefix="xxx"
```

Where the 'xxx' can be any alphanumeric characters. But whatever the prefix, it must be followed by 4 digits to represent the partition number (0000, 0001, 0002, etc.).

Ok, so what prefix do I want? It needs to be short, because with the additional 4 digits, and a limit of 16 characters before the OLED begins scrolling, I still wanted space to show a meaningful filename.

Gotek Root Directory File Name Strategy

As we have discussed a few times already, before using a USB flash drive in the Gotek, it must be prepared on a host computer, such as a PC-clone. Plug the USB flash drive into an available USB port and after ensuring it is formatted with a FAT system, place the following files in the Root directory:

- FF.CFG Configures the Gotek for system attributes, such as OLED, sound, Rotary Encoder attributes and Navigation Image Mode.
- IMG.CFG Configured for the specific type of drive being emulated (Optional for a standard 1.44Mb 3" drive).
- IMAGE_A.CFG File generated by the Gotek as the last partition accessed.
- DSKA0000.hfe Size-specific blank image files for each of the desired partitions to be used.

The above Image File Name was adequate for the 3-digit, 7-segment display of the normal Gotek (please see the earlier Gotek articles on my *Z-100 LifeLine* website for more information), but if you have an OLED display, a bit more information is very helpful. The number of characters displayed on the Gotek OLED display is 16 before it begins to scroll. As I don't like the scrolling display, I try to keep the file names less than 16 characters.

If you decide to change a file name, it will NOT change the contents of that particular partition. However, in the Native Image Mode, you have no control over the actual order the file names will appear in and what partition numbers are used (except that the number of image files will determine the maximum number of usable partitions).

Note: You must include a blank image file for each, and every, partition that you wish to use. Ten image files will give you 10 partitions; 100 will give you 100 partitions. You can leave partitions blank (unused) for future use by including blank partition names.

Before we continue, we must address one more consideration - What size drive do we wish to emulate? What would be most useful and meet all the requirements we discussed last time regarding floppy disk size?

Well, my hard drive (MFM, 10Mb) has 4 partitions and is able to run ZDOS v3, ZDOS v4, PC-DOS (with Gemini board), and CP/M. Would it not be best to be able to store and run all these files from one floppy disk system? And what could be better than a high capacity 8" drive? Last time we discussed all the various limitations that the various operating systems place on floppy drive systems, but the one drive system seen by all is the lowly 8" drive. No DiskPack requirements. No real cabling difficulties - just a 34-pin cable to 50-pin interface cable.

Oh, and while I'm on the subject, the Gotek IS CAPABLE of sending a Ready Signal on pin 34. Another FF.CFG setting!

```
# Pins 2 & 34 output (drive->host) manual configuration
# auto: Auto-configure from interface= setting
# nc: Unused/No Connection [eg. if pin is an input (host->drive) on your host]
# low: Always 0 volts (0v)
# high: Always 5 volts (5v)
# rdy: Drive ready (Ready = 0v)
# nrdy: Logical complement of above
# dens: Density mode (High Density = 0v)
# ndens: Logical complement of above
# chg: Disk changed (Changed = 0v)
# nchg: Logical complement of above
# Values: auto, nc, low, high, rdy, nrdy, dens, ndens, chg, nchg
pin02 = nc
pin34 = rdy
```

Important Note: Setting pin 34 to RDY in FF.CFG does indeed work on my Gotek and I could remove the diodes in my 50-pin cable and also remove the useless JC jumper on the Gotek. Just go with the basic 34-pin to 50-pin adapter cable, as discussed last time.

Note: There is one last consideration - PC use? The newest PC computers may have an issue with the use of any disk emulator that does not show a 3", 1.44Mb drive as drive A:. I do not recall ANY PC that permitted attaching an 8" drive, even though the DOS software of the period mentions 8" (we will see this later).

Anyway, you talked me into it. I'll go with the 8" drive emulation. Here's the system data that you would need...

Double Density 8" 1.2Mb Emulation

The 8" 1.2Mb double density disk system is nearly identical to the High Density 5", 96 tpi, 1.2Mb disk system, but only uses 77 tracks. So, while the HxC software does not provide us with 8" blank image files, you can use the same ones as the HD 5", 1.2Mb disks.

The 8" 1.2Mb double density drive system requires the special configuration files, IMG.CFG and FF.CFG, in addition to their own set of blank disk image files in the Root directory of the USB flash drive.

Note: The blank image disk files MUST match the disk size specified in the IMG.CFG file. This must be done before placement of the USB drive in the Gotek Emulator.

Add a third line to the IMG.CFG file to reflect the drive size:

```
## Set for 8", 48 tpi, DoubleDen, 360rpm, 77 track, 1.2Mb drive.
```

I changed the IMG.CFG file to reflect the 8" drives, as follows:

```
# [dsdd80::1261568] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 1232kB (1261568 bytes).
[dsdd80::1261568]

# Mandatory: Number of cylinders (1-255).
cyls = 77

# Number of sectors per track (0-256). Default is 0.
secs = 8

# Bytes per sector (128, 256, 512, 1024, 2048, 4096, 8192).
# Mandatory if @secs is non-zero.
bps = 1024

# Rotational RPM. Default is 300.
rpm = 360
```

Note: While in the PC, any image files are placed in alphabetical order. However, in the Native Image Navigation Mode, the Gotek appears to rearrange file names in some other unpredictable order and any new file images are added at the end of or after any existing partitions. This makes it difficult to maintain most kinds of file index, or maintaining files in any kind of groupings - such as all ZDOS v4 files or a group of CP/M files.

For example, I attempted to arrange the initial set of files using the PC to order them alphabetically, hoping the Gotek would use the files in the order listed below. I wanted to identify the emulated disk size with the first number (for example 8 to denote an 8" drive) the next 4 digits to list the partition number, then a file name describing the operating system and contents, such as in the following form:

```
80000_(VolName).hfe
```

Example File Names for an 8" Drive Strategy:

```
80000_DOS4_Boot.hfe
80001_DOS4_BkUp.hfe
80002_DOS4_DIAG.hfe
80003_DOS4_.hfe (future use)
80004_DOS4_.hfe (future use)
...
```

```
80010_WordStar.hfe
80011_WS_.hfe (future use)
80012_WS_.hfe (future use)
...
80020_SuperCalc.hfe
80021_SC_.hfe (future use)
80022_SC_.hfe (future use)
...
80100_DOS3_Boot.hfe, for ZDOS v3
80101_DOS3_BkUp.hfe, for ZDOS v3
80111_.hfe (future use)
80112_.hfe (future use)
...
80800_85_Boot.hfe, for CP/M-85
80801_85_BkUp.hfe, for CP/M-85
80802_85_.hfe (future use)
80813_85_.hfe (future use)
```

In the Native Navigation Mode, after the first few, these were seemingly placed in random partitions (probably placed in order of their creation), such as what happens during a COPY operation in the Z-100. There was no way that this was going to work.

I tried the Indexed Navigation Mode next. However, you need to make changes to the FF.CFG file:

```
# Navigation mode for selecting images/slots
# native: Navigate through all valid images/dirs
# indexed: Navigate through DSKA0000, DSKA0001, ...
# default: native unless overridden by HxC-compat-mode config
nav-mode = indexed

# Prefix for image names (DSKA0000.hfe, DSKA0001.hfe, etc.)
# in indexed navigation mode. Default prefix = "DSKA".
# Must use 4 digits after prefix. String can be empty ("").
indexed-prefix = "8"
```

This Indexed Mode does precisely what I had been searching for and it works as I had wanted all along. I backed up my entire hard drive to the Gotek with 8" disk emulation. The names show nicely on the OLED and the buttons work great. The only pain is having to place the USB flash drive into the PC to add or update partitions. Changing the partition names makes no changes to the partition contents, however, if you change the partition number in the name, the reported partition (and its contents) move with it.

Important Note: While using ZDOSv3, on a ZBASIC partition where I did not format the partition, when I tried to copy all the files from a ZBASIC Demo disk, most of the files did not appear in the directory, though the space available did not match with what was reportedly used. The Demo disk had all kinds of files with various extensions, such as .DAT, .FIL, etc., but the only files that appeared in the directory listing had the .BAS and a few files with other common extensions. After I ran FORMAT /v on the partition, the files copied fine. Weird.

So, I recommend you get in the habit of always running FORMAT on all the Gotek partitions as you use them.

It was amazing how much space 1.2Mb on a Z-100 is. I can boot many versions of ZDOS and two versions of CP/M on the same 8" Gotek emulator.

However, I worked for days trying to get PC-DOS to work on a Gemini PC emulator board. Here's that story...

PC-DOS and the Gemini PC-Emulator

The Z-100 series computers can emulate an IBM PC-Clone with the use of a Gemini or EasyPC PC-emulator system (with a few limitations), or a software PC-emulation package called ZPC (again with several limitations). The Scottie Board, specifically designed to work with ZPC is also somewhat successful, but I have never used it. There are many articles in my periodical index (check out the index on the *Z-100 LifeLine* website) that describe these PC-emulation systems.

However, some versions of Zenith's MS-DOS designed for the Zenith PC computers must be run under the Z-100 series computer's Gemini or EasyPC PC-emulator systems. I'm using Zenith's MS-DOS v3.21 which I refer to as PC-DOS, to separate it from all the other versions of DOS out there. So, to me, there are only PC-DOS and ZDOS.

Anyway, after several days of effort, I can NOT get the 8" Gotek to work with PC-DOS. The problem could be from PC-DOS, the Gemini PC-emulator system, or me, but here is how it went...

First I tried attaching the 8" Gotek to the 50-pin interface of the Z-207 Floppy Disk Controller. However, you must remember that Zenith's PC-DOS was designed for their PC-clone computers that did not have a separate interface for high density drives. No matter what I tried, I could not get the Gotek recognized on the 50-pin interface.

So, I figured that the Gotek will have to appear on the 34-pin interface, as unit 1.

By default, this version of PC-DOS recognizes drives A: and B: as floppy drives, and any existing hard drive partitions as drives C:, D:, E: and F:. But there are two means of adding additional drivers to add other drives, tape drives, and other block devices:

The first I tried was DRIVPARM:

According to the PC-DOS manual, DRIVPARM defines parameters for block devices (such as floppy disk drives, tape drives, etc.) when you start PC-DOS, overriding the original DOS device driver settings.

Syntax:

```
DRIVPARM= /d:dd [/c] [[/f:ff] [/h:hh] [/n] [/s:ss] [/t:tt]
```

Where:

/d:dd Specifies a **logical** drive number between 0 and 255. This means that drive number 0=A, 1=B, 2=C, and so on.

/c Specifies that change-line (doorlock) support is required.

/f:ff Specifies the form factor index, where:

- 0 = 320/360K bytes
- 1 = 1.2M bytes
- 2 = 720K bytes
- 3 = 8" Single Density
- 4 = 8" Double Density
- 5 = Hard Disk
- 6 = Tape Drive
- 7 = Other

Note: Except for the 1.2Mb for a 5" 96 tpi drive (or possibly an 8" 1.2Mb drive), there are NO choices for a HD, 1.44Mb, 3" drive!

The default values for the following parameters depend upon the form factor specified with the **/f:** switch. If you do not specify the **/f:** switch, **drivparm** uses a default of 720K bytes.

/h:hh Specifies the maximum head number (From 1 to 99).

/n Specifies a nonremovable block device.

/s:ss Specifies the number of sectors per track (From 1 to 99).

/t:ttt Specifies the number of tracks per side on the block device (From 1 to 999).

THE ONLY place that the 8" Gotek floppy disk emulator was recognized at all was as drive unit 1, on the 34-pin Z207 interface, with the following line added to CONFIG.SYS:

```
drivparm=/d:1 /f:4 /h:2 /s:8 /t:77
```

The Gotek is recognized as drive B:, but if you run FORMAT B:/s/v:

- The initial 40 tracks seem to format as usual (fairly rapid pace similar to a normal drive).
- After track 40, each head (0 and 1) is run for about a full minute or more before proceeding to the NEXT track, while the Gotek OLED is displaying the correct track number and head for a bit. It then changes to T99.0 or T99.1 for most of the time and an occasional 'Direct Access' appears on the top line shortly before the Txx.x display goes back to the correct track number, and the track finally increments.
- More than an hour later, the track numbers continued to increase to somewhere near track 88 (I did not catch the exact Track), the format was completed successfully, and the system files were copied. After the usual request for a volume name, the last step displays:

```
"System Transferred"  
"730112 Bytes total disk space"  
" 72704 Bytes used by system"  
"225280 Bytes in BAD Sectors!"  
"432128 Bytes available on disk"!!
```
- Obviously, the delay was caused by testing BAD sectors!
- And the disk size, 720Kb, was wrong!
- Changing the Form Factor to **/f:1 = 1.2M bytes** had no change from **/f:4 = 8"DD**.

Another command that could be used in CONFIG.SYS was DRIVER.SYS, with a similar description:

DRIVER.SYS is an installable device driver that supports external drives.

Note: If you want to configure a logical device, use the DRIVPARM command described earlier.

To install DRIVER.SYS, you must use the following syntax for the command line in your CONFIG.SYS file:

```
DEVICE=DRIVER.SYS /d:dd [/c] [[/f:ff] [/h:hh] [/n] [/s:ss] [/t:tt]
```

Where:

/d:dd Specifies a **physical** drive number between 0 and 255, but these are numbered differently from logical drives. Floppy drives are numbered starting at 0.

For example: If you have two floppy drives, they might be numbered 0 and 1. If you were to add an external floppy drive, its physical drive number would be 2. However, if you only have one floppy drive, numbered 0, if you add an external floppy drive, its physical drive number would still be 2, because MS-DOS already knows the definitions of physical drives 0 and 1.

So, for my Gotek, I'm limited to **/d:2**.

/c Specifies that change-line (doorlock) support is required.
/f:ff Specifies the form factor index, where:

0 = 320/360K bytes
1 = 1.2M bytes
2 = 720K bytes
3 = 8" Single Density
4 = 8" Double Density
5 = Hard Disk
6 = Tape Drive
7 = Other

Note: Except for the 1.2Mb for a 5" 96 tpi drive (or possibly an 8" 1.2Mb drive), there are NO choices for a HD, 1.44Mb, 3" drive!

The default values for the following parameters depend upon the form factor specified with the **/f:** switch. If you do not specify the **/f:** switch, **DRIVER.SYS** uses a default of 720K bytes.

/h:hh Specifies the maximum head number (From 1 to 99).
/n Specifies a nonremovable block device.
/s:ss Specifies the number of sectors per track (From 1 to 99).
/t:ttt Specifies the number of tracks per side on the block device (From 1 to 999).

For example: If you want to add an external 720K byte drive to the computer, you would include the following line in the CONFIG.SYS file:

```
device=DRIVER.SYS /d:02
```

DRIVER.SYS can also be used to provide *alias* drive letters for internal drives that already have assigned letters. For example, if drive A is a 1.2Mb floppy disk drive and you include:

```
device=DRIVER.SYS /d:0 /f:1 /h:2 /c /s:15 /t:80
```

in your CONFIG.SYS file, your first floppy drive (drive A) would have two drive letters (the default of A and - assuming you have only one hard disk - letter D, because of DRIVER.SYS).

I tried the command:

```
device=DRIVER.SYS /d:2 /f:4 /h:2 /s:8 /t:77
```

but no matter what I tried, the Gotek could not be recognized. I tried **/f:1** also to no avail. I even configured the Gotek as drive unit 0 on a cable with a twist on the 34-pin connector, making it unit 2, but no drive LED.

I thought about making the Gotek emulate 3" or 5", but while the 5" might work if the Z-100 is configured to run 5" 96 tpi real drives, it would not be worth the bother to convert everything to 5" HD disks.

While the Gemini PC emulator may be the problem, I would not expect any more success running the EasyPC emulator system. Zenith's PC-DOS was created for running PC-DOS on Zenith PC-clones and these did not have floppy controllers similar to that of the Z-100. So, PC-DOS knows nothing of the Z-100's 50-pin interface for handling High Density drives, and nothing connected to the 50-pin interface was recognized.

Further, I had only limited success with the Gotek on the 34-pin connector as drive B:, and then it was low density.

Sorry. I see no further ways to proceed.

If you have some success, I would love to hear about it.

Creation of Image Files

Further research has shown that the files placed into a Gotek partition while installed on the Z-100 series computer are indeed saved in the image file that we discussed earlier. If you look at the files in the Root Directory of the USB flash drive in the host computer, you can see that all the image files appear unchanged from when you initially installed them - same date and size as first loaded.

However, those image files contain the exact contents that were loaded into each partition while the Gotek was operating on the Z-100.

This caught me by surprise when I was preparing a second 8" USB flash drive, and when I put this second, new USB drive in the Gotek and searched it in the Z-100, all the files were there in the new USB flash drive!

To confirm this I checked the contents of several of the supposedly blank file names in the USB on the PC, using the HxC File Browser and the files were there.

Let's review how these can be seen.

You may recall when we were creating the blank image files for the USB flash Drive in our PC (or other host computer), we used some HxC software to create them. Let me explain it again.

Insert your USB flash drive in a spare USB port on your PC or other host computer.

My HxC software was downloaded and unZIPPed to my Downloads folder. To invoke the HxCFloppyEmulator.EXE file, I need to follow the sequence:

```
Downloads > HxCFloppyEmulator_Soft > HxCFloppyEmulator  
_Software > Windows > HxCFloppyEmulator.EXE
```

Depending upon where you placed your downloaded HxC software, your path will be different.

Double-Click on the HxCFloppyEmulator.EXE file and a window opens on the screen, showing "HxCFloppyEmulator v2.5.6.6" and a screen of options.

To view the contents of your Image files, select [Disk Browser]. A new window will open, "DOS Floppy Disk File Browser", which shows the File system type:

```
3"5 720KB DSDD FAT12 (as a default)
```

and a blank screen below it. To load and see an image file, click on [Load Image] on the right side and browse to where you store your image files in the PC (or your host computer) or the USB flash drive in one of the USB ports.

Double-click or select the desired file and the contents will appear in the window, complete with a directory structure. The partition data is shown in the bottom line.

Note: This is a DOS File Browser. It will not work on image files created under other operating systems. For example, the Disk Browser does NOT recognize ZDOS3 image files. It gives an error:

```
"Null sized DOS image! Invalid Disk!"
```

However, the image is still there and still works. It was just created while operating under ZDOS v3. The Browser also does NOT recognize CP/M disks that were created while operating under CP/M, giving the error:

```
"Some error found during the image mounting! DOS Disk?
Disk Damaged!"
```

You would need to create a disk image of the CP/M disk using imaging software, such as this HxC software - if it can read CP/M disks.

All this being said, the easiest way to backup any USB flash drive, would be to copy the image files from the Gotek's USB flash drive back into your host computer for storage. Then, if you need another USB flash drive of the **same floppy size**, just copy the saved image files to the new USB flash drive. Job done.

Oh, by the way, I got tired of squinting to read the default display so I changed the screen size to 8x16 in the FF.CFG file. I also found that the screen timed out too fast for me, so I also changed the timeout to 200. I found these settings work better and the names can be up to 16 characters long before the screen scrolls.

Configure OLED Display Size in FF.CFG.

```
# OLED Font. Narrow and wide options.
# Narrower 6x13 font permits:
# - More characters per row
# - Use of Gotek display cutout
# (eg. "display-type=oled-128x32-narrow")
# Values: 6x13 | 8x16
oled-font = 8x16
```

Configure OLED Display Off in FF.CFG

```
# Turn an LCD or OLED display off after N seconds of inactivity
# N=0: always off; N=255: always on
# Values: 0 <= N <= 255
display-off-secs = 240
```

Closing:

It bears repeating again. **You do not need to create a disk image, to back up a disk.** Gotek already creates the disk image of any files that you copy to a partition in the Gotek, and are in the size of the disk emulation chosen for the USB flash drive.

When you have copied all your files to the USB flash drive in the Gotek, place the USB flash drive back into your host computer and save the files in the USB flash drive's Root directory into a folder in the host computer. Then, any time you need to create another USB flash drive, OF THAT SAME SIZE, just place the image files in the Root directory of the new USB flash drive. All the files loaded on the original USB flash drive will then appear in the new USB flash drive!

Note: You cannot copy the file images of one floppy size to those of another floppy size. For example, you can not copy the file image of an 8" floppy to create the file image of a 3" floppy. The file copied will still appear as an 8" floppy and will, in most cases, generate an error.

I have not addressed the procedures used to create disk images of your old software using other imaging software, as it is outside the scope of my articles and the procedures will vary among software packages. However, the HxC software that I used to create my blank disk images appears to have all the capability that you may need. I would begin there. There is also a lot of information at the FlashFloppy and the HxC websites, and other on-line sites to assist you.

Using blank images, you can create the number of partitions needed, or add them at any time by just adding additional blank disk images anytime in the future. I do recommend retaining a disk drive of each type for future use.

Note: I'll update this manual with new information as it becomes available or changes. If you wish to add any additional information and/or your comments on experimentation, please let me know at: z100lifeline@swvagts.com

I hope this helps clarify the use of the Gotek Floppy Emulator in the Z-100 series computer.

Stay safe.

Cheers,



Steven W. Vagts
Editor, Z-100 LifeLine
z100lifeline@swvagts.com

