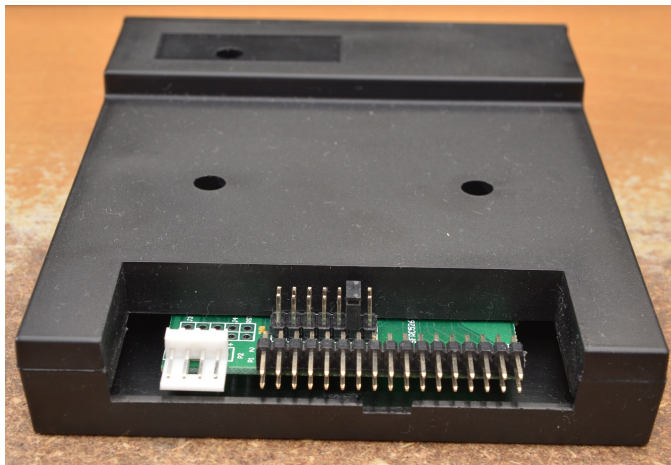




Gotek Floppy Emulator Firmware Upgrade



Note: During normal operation, pressing both buttons for any length of time will NOT cause a reformat of the USB flash drive; it just causes the hundredths digit to increment. To reformat the USB flash drive, you must press both buttons while turning power ON.



GOTEK Floppy Emulator

In the last issue we discussed the capabilities, cabling requirements, and limitations of the basic Gotek Floppy Disk Emulator, as bought, in depth, 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

Gotek Floppy Drive Emulator Features

- Uses normal USB flash drive as the medium for storage design.
- Has 34-pin interface and 5Vdc standard 3.5" power plug, easy to install.
- Has 1.44Mb Storage Capacity per Partition, formatted as:
 - Cylinders: 80
 - Tracks (both sides): 160
 - Sectors per Track: 18
 - Bytes/Sector: 512
 - Data Transfer Rate: 500 kbps
 - Encoding Method: MFM
- Can use 1000 Partitions maximum in a 2Gb USB flash drive (or larger). Smaller flash drives have fewer partitions for use.
- Built-in self-format function by turning power ON while pressing both buttons until digits begin to increment from 000.
- Jumpers with function for different machines (The SFR1M44-U100K does NOT support all jumpers)
- Has three 7-segment numbers to represent partition number (000-999)
- Green LED: Status of read and write operation.
- Two buttons (Increment only):
 - Right button increments number single-digit
 - Left button increments number tens-digit
 - Both buttons together increments hundreds-digit

View of Rear Panel:

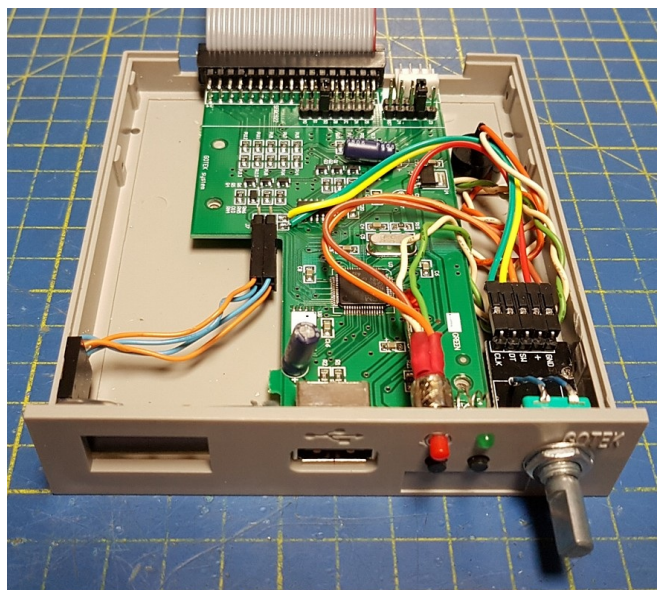
- Has 34-pin drive connector interface
- Has standard 3.5" 5Vdc_GND(2)_12Vdc power plug
- Thru-holes to add programming jumpers
- Normal Operation Jumper Settings:

	SFR1M44-TU100K	SFR1M44-U100K-R	SFR1M44-U100K
J5:	---	---	---
JA:	Set to generate READY signal on 34-pin Floppy Drive Interface		
JC:	---	---	---
JB:	---	---	---
S0:	Drive Select 0		Drive Select 0
S1:	Drive Select 1		Drive Select 1
MO:	Motor On		Motor On

FlashFloppy Gotek Course 201

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

Probably, the most critical upgrade that you can make to the Gotek Floppy Disk Emulator is to the firmware. The original firmware has limited capability and the aftermarket firmware upgrades are considerably more powerful, adding capability and flexibility to the original product.



Interior of an Upgraded Gotek

Presently, there are two very similar firmware upgrades available that promise several improvements over the standard 1.44 Mb 3" drive emulator; the FlashFloppy Firmware upgrade and the HxC Firmware upgrade. Each supports hundreds of computers, keyboards, music synthesizers, and computer controlled machines, such as textile or milling equipment. This article will discuss the FlashFloppy firmware upgrade because it is free. But, both support:

- 3.5", 5.25" (48 tpi and 96 tpi), and 8" disks
- Single and Double sided disks
- FM / SD, MFM / DD or HD densities
- 125Kb/s, 250Kb/s, 300Kb/s, 500Kb/s Bitrates
- 300 RPM, 360 RPM, 600 RPM
- 128/256/512/1024/2048/4096 Byte sector sizes
- All track skew and sector interleaving
- Up to 255 tracks per disk
- And much more!

Software Downloads for Preparation of the Gotek

We will require software from several websites to be able to program the new firmware update of our Gotek and to prepare a USB flash drive for use on our particular Z-100 computer systems. So before we start, let's download all that we will need, and make sure it is still available.

Download the following software to your PC or other computer host from which you will update the firmware on your Gotek and will prepare your USB flash drive.

Note: Since this article was published (March 2021), software or its name and purpose may have changed, been updated, or even removed. If a website or the software can no longer be found, please let me know and I may be able to assist you with what you are missing.

The website:

<https://www.st.com/en/development-tools/flasher-stm32.html>

has the software necessary to flash our firmware using Method 1- Serial Programming (see below). If you choose this method, you must download and use the STM32 Flash Loader Demonstrator (UM0462) from this site. The filename is: **en.flashier-stm32.zip**.

The website:

<https://www.st.com/en/development-tools/stsw-stm32080.html>

has the software necessary to flash our firmware using Method 2 - USB Programming (see below). If you choose this method, you must download and use the DfuSe USB device firmware upgrade (UM0412).

You may need to obtain a software driver for your host computer (PC or other) for whatever model of USB to serial (TTL) device you intend to use to flash the Gotek. I had purchased the CH340G USB to TTL UART module (Method 1). Searching online found the site:

<https://roboindia.com/tutorials/ch340-usb-to-serial-driver-installation/>

and I could download the driver: **CH341SER.ZIP** from there.

You will need the Firmware Upgrade itself. As we are using FlashFloppy for this article, you will need to visit the site:

<https://github.com/keirf/FlashFloppy/wiki/Firmware-Update/>

to download the latest Firmware; **flashfloppy-v3.23.zip** or later. To prepare the USB flash drive, you will also need the software to place in the device's Root directory. At this same site, under "Usage" in the right column, check out both "FF.CFG Configuration File" and the "IMG.CFG Configuration File". Both these files are in the FlashFloppy software in the "Examples" folder. We will discuss these in detail later.

Finally, we will need some disk image files to place in the USB flash drive's Root directory. As I did not see the FlashFloppy versions, I used HxC's Floppy Emulator Site:

https://hxc2001.com/download/floppy_drive_emulator/

and clicked on "USB Stick STM32 HxC Floppy Emulator (HxC Firmware for Gotek)". Under "Ready to Use File Images" I clicked on:

"QuickInstall_FloppyDiskImages.zip".

And while we are here, I found the HxCFloppyEmulator Software very handy in creating or modifying these disk images. A few lines further down, under "HxCFloppyEmulator Software", click on and download:

"HxCFloppyEmulator_soft.zip".

There is a lot of interesting information to be found at these last two sites. Take some time to review all that these sites have to offer. As we discuss the following procedures, they may be helpful in clarifying what I will be covering. I suggest marking both as favorites.

Hardware Preparation of the Gotek

In order to upgrade the Gotek firmware, you will need to install some jumper pins to the holes located at the rear of the circuit board near the existing jumper pins. The number of pins vary, mine had a row of five and a row of four. Some models had only two in the second row.

Note: While you are installing pins, your Gotek model might also need additional pins installed for the set of pins used for the Rotary Encoder

upgrade. My model only had the four pins for the existing display and two soldered over holes for a third pair to be installed. I removed the existing 90 degree header with four pins and installed another 90 degree header of six pins. However, read about my experiences with the rotary encoder in the accompanying Hardware Upgrade article before making a decision on adding pins here.

The first installation of this FlashFloppy firmware to the Gotek can be done either by serial or USB link from a host PC.

Method 1: Serial Programming:

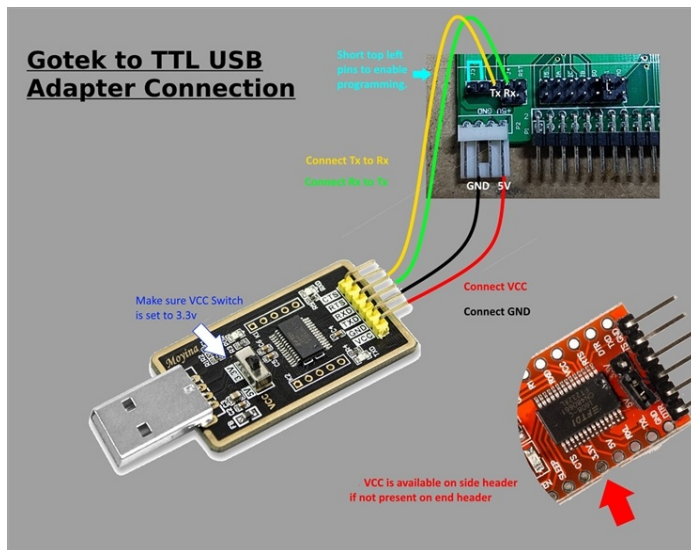
You will also need a USB serial adapter or USB TTL adapter to install the firmware from your PC to the Gotek Emulator. There are many different models, but just look for a model that has a USB port and a minimum of four pins; +5V, Ground, RX and TX. On mine, the pins were labeled: 5V, GND, RXD and TXD. It also had a pin labeled 3V3, which will not be used for the Gotek Firmware Upgrade.

To connect the USB adapter to the Gotek Emulator, refer to the following diagram and attach:

- 5V pin of the adapter to the 5V pin on the power connector of the Gotek drive.
- Gnd pin of the adapter to the GND pin (either of the center two pins) on the power connector.
- TXD pin to the RX pin of the new pins we just installed as the program plug on the Gotek. The Rx pin is next to the label, J4.
- RXD pin to the TX pin of the program plug on the Gotek.
- Short the left two pins, labeled J3, on the program plug with a jumper to enable programming.

Note: The ordering of the connections (5V,GND,TX,RX) can vary across USB-TTL serial adapters; be careful to make the correct connections!

There are also two pins remaining at the right end of the programming plug in the diagram. One is labeled RST and the other is in front of it in the diagram. These are the reset pins. Short these briefly to reset the programming, but only if the computer asks for this to be done.



Note: This diagram shows selecting 3.3V on the adapter, though the picture shows the switch in the 5 volt position. My adapter board had a jumper selection. Some of the videos that I've seen use an adapter

with no such voltage selection. In any case, if you have a selection, try 3.3V first. If the programming does not work, try the 5V selection.

Note: Some serial adapters may require pull-up resistors on the TX and RX wires. A suitable resistor value (R) is 10K; these can safely be connected to VCC/+V of 3.3V or 5V.

The Flash Chip, from ST Microelectronics, on the Gotek Emulator circuit board is labeled:

ARMz
STM32F105
RBT6
991XD

The website:

<https://www.st.com/en/development-tools/flasher-stm32.html>

has the software necessary to flash our firmware. You need to download and use the STM32 Flash Loader Demonstrator (UM0462).

The STM32 Flash Loader Demonstrator (FLASHER-STM32) is a free software PC utility from STMicroelectronics, which runs on Microsoft Operating Systems and communicates through the computer's USB RS232 with the STM32 microcontroller system memory boot mode.

Note: This software utility also contains a command line version and it is provided with Microsoft Visual Studio 12 source code. The STM32 Flash Loader Demonstrator is available for other operating systems.

While the software is free, when you download it, you will need to accept a licensing agreement and provide a name and email address. Click on the Download button to download the en.flasher-stm32.zip file. Unzip to flash_loader_demo_v2.8.0.exe or later.

- With the Gotek attached properly to your Flash Adapter, using a USB cable, connect your Flash Adapter to the computer.
- Once downloaded, double-click on Flash_loader_demo_v2.8.0.exe to invoke the program.

That will activate the FlashLoader Demonstrator 2.8.0 - InstallShield Wizard:

Welcome to the InstallShield Wizard for FlashLoader Demonstrator 2.8.0
The InstallShield Wizard will install FlashLoader Demonstrator 2.8.0 on your computer. To continue, click next.

[Next]

I clicked [Next] and the installer continued with a new window:

Customer Information

Please enter your information

User Name: [z100lifeline@swvagts.com]

Organization: [Z-100 LifeLine]

[<Back]

[Next >]

[Cancel]

I filled in the requested information and clicked [Next]. Another window appeared:

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install FlashLoader Demonstrator 2.8.0 to:
 C:\Program Files (x86)\STMicroelectronics\ [Change]

[<Back] [Next >] [Cancel]

I accepted the folder and clicked [Next].
 Another window:

Ready to Install the Program
 The Wizard is ready to begin installation

If you want to review or change any of your installation settings, click Back.
 Click Cancel to exit the wizard.

Current Settings:

Setup Type: Typical
 Destination Folder:
 [C:\Program Files (x86)\STMicroelectronics\]
 User Information:
 Name: z100lifeline@swvagts.com
 Company: Z-100 LifeLine

[<Back] [Install] [Cancel]

Clicking [Install], the computer flashes through other windows and brings you to:

InstallShield Wizard Completed
 The InstallShield Wizard has successfully installed FlashLoader Demonstrator 2.8.0.
 Click Finish to exit the wizard.

[X] Launch the program
 [] Show the readme file (not very helpful)

[Finish]

The Flash Loader Demonstrator opens with:

Select the communication port and set settings, then click next to open connection.

Common for all families:
 (*) UART

Port Name	[COM3]	Parity	[Even]
Baud Rate	[115200]	Echo	[Disabled]
Data Bits	[8]	Timeout(s)	[10]

[Next] [Close]

You are supposed to type in the Port Name (shown above) and change any other values, if needed. But it would not let me type in a Port Name, so I removed and reinserted the programming adapter into the USB port. Windows 10 immediately recognized the new device and after a minute, it said it was ready. But nothing showed up for a Port Name. I clicked [Next], but in another window I got the error:

"Cannot open the COM port, the port may be used by another application."

[OK]

I clicked [OK], which only closed that window and left me with the previous window, waiting for a COM port.

Searching the Device Manager in Control Panel, I found that the USB2.0-Ser! did **not** have a driver and could not locate one! But further research in Device Manager showed that this idiot computer did not list any Ports (COM & LPT) either!

I had to click on 'Action' in the task bar, which initially showed only a 'Help' option. But uninstalling the USB2.0-Ser! device brought up a few more options. Clicking on 'Add Legacy Hardware', then on 'Communications Port', this added the 'Ports (COM & LPT)' line that I needed in Device Manager, but there was an exclamation mark on 'Communications Port (COM3)', because the device did not have any settings that it needed. I had to [close] and reboot.

To make a very long process as short as possible, I did not need to worry about ports. Rather, I found the model of my USB2.0 to serial (TTL) device from my Ebay order; I had the CH340G USB to TTL UART module. Searching online found the site:

<https://roboindia.com/tutorials/ch340-usb-to-serial-driver-installation/>

and I could download the driver: CH341SER.ZIP from there. I downloaded and unzipped the file, then double-clicked to open the file: **SETUP.EXE**

A window opened showing:

DriversSetup(x64)
 Device driver install/uninstall
 Select INF [CH341SER.INF]
 [Install]
 [Uninstall]

Clicking on [Install] the first time brought up a new window:

DriversSetup Driver install failed

But I plugged the USB adapter into a USB port on the PC; tried again:

DriversSetup Driver install success!

Reentering Control Panel and Device Manager I found a new line under Ports (COM & LPT):

USB-serial CH340 (COM4)

and I could remove the 'Communications Port (COM3)' with the exclamation mark.

Clicking on the Windows Start Button and hovering over 'Programs' > 'STMicroelectronics' > to click on 'Flashloader', brought me back to the window where I left off:

Select the communication port and set settings,
 then click next to open connection.

Common for all families:
 (*) UART

Port Name	[COM4]	Parity	[Even]
Baud Rate	[115200]	Echo	[Disabled]
Data Bits	[8]	Timeout(s)	[10]

[Next] [Close]

With COM4 now already inserted, I clicked [Next], but an error popped up:

STMPFlashLoader Demo
Unrecognized device - Please reset your device, then try again

[OK]

This is when you will need to short the RST pins briefly, as I described earlier. Click [OK], and if everything is good, you get the warning (with a red stop light):

WARNING: When you click "Remove Protection" the flash will be mass erased and all data will be lost.

[Remove Protection]

[Back] [Close]

Clicking on [Remove Protection] gave the message:

Wait until full erase

After about a minute, a green stoplight appeared with:

Target is readable. Please click "Next" to proceed.
Flash size 128Kb

[Back] [Next] [Close]

Clicking [Next] brought up the next window, with a green stoplight:

Flash Loader Demonstrator

Target is readable. Please click "Next" to proceed.
Flash Size 128Kb

[Back] [Next] [Close]

Click [Next], and the program continued with:

Flash Loader Demonstrator"

Please select your device in the target list
Target [STM32F1_Connectivityline_128K]
(Filled in automatically)
RID (h) [0418]
BID (h) [NA]
Version [2.2]

Flash mapping:

Name: Start Address: End Address: Size: RW
(A List of Pages is shown)
... Page0
... Page1
... Page2
... Etc.

Legend: [Red lock] Protected [Green lock] Unprotected

[Back] [Next] [Close]

Installing the FlashFloppy Firmware

Now it is time to choose the firmware to install. We shall use the Flash Floppy Firmware, described next. Once you have downloaded FlashFloppy-v3.16.zip (or later), unzip the file.

We only need the file, FF_Gotek-v3.16.hex, so save this file separately to the download directory or another handy location on the PC.

Click [Next] from the previous screen and we get another page:

Flash Loader Demonstrator

() Erase (*) All () Selection

(*) Download to device

Download from file
[C:\...\Downloads\flashfloppy-v3.23\FF_Gotek-v3.16.hex]

(*) Erase necessary pages () No erase () Global Erase
"@ (h) [8000000] [] Jump to the user program
[] Optimize (Remove some FFs) [] Verify after download
[] Apply option bytes []

() Upload from device
Upload to file []

() Enable/Disable Flash Protection
[DISABLE] [WRITE PROTECTION]

() Edit Option Bytes

[Back] [Next] [Close]

Click on [Next] and the next screen shows:

Flash Loader Demonstrator

Target STM32F1_connectivityline_128K
Map file STM32F1_connectivityline_128K.STmap

Operation DOWNLOAD
File name C:\...\Downloads\flashfloppy-v3.23\FF_Gotek-v3.16.hex

File size 109.96KB (112596 bytes)
Status xxx.xxB (112596 bytes) of 109.96 KB (112596 bytes)
Time 00:xx

[Erasing flash 00%]

Where xx's update in value as the programming continues and the progress bar changes to:

[Downloading data xx%]
(turns blue as progress is made)

[Cancel]

The bar turns green when the operation is complete:

[Download operation finished successfully]

[Back] [Close]

Click [Close]. The update is complete, and the Gotek drive can now be checked.

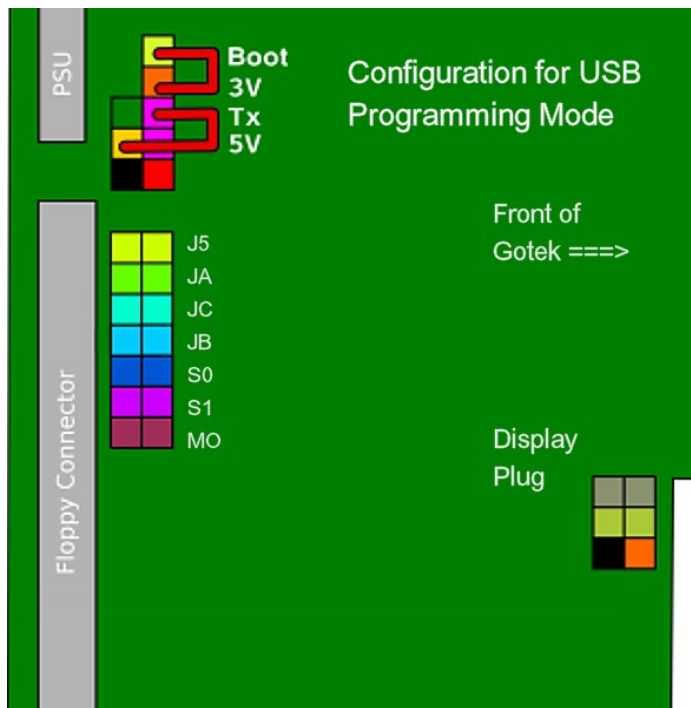
You can also begin any hardware upgrades you wish to make, but I recommend you try the Gotek first, get some experience in its use and ensure it is working as it should.

Method 2: USB Programming:

This method requires a USB-A to USB-A cable, a pair of jumper wires to configure the programming header of the Gotek, and the *.dfu programming file contained in the FlashFloppy distribution folder.

Note: At the time of this publication, this USB method cannot be used to upgrade to HxC Firmware because HxC uses the USB to serial (TTL) adapter as a form of copy protection.

To configure the programming header of the Gotek for USB programming, temporarily place jumper wires across the following pin pairs: Boot - 3V and Tx - 5V.



Note: Because these jumper wires are temporary and only used for this initial programming, you can simply use short lengths of hookup wire (let the spring tension in the wire keep them in place). Just ensure the bare ends do not short out.

Connect the Gotek to a USB port your host using a USB-A male to USB-A male cable. Use the appropriate *.dfu file included in the Flash Floppy distribution folder, when called for.

The website:

<https://www.st.com/en/development-tools/stsw-stm32080.html>

has the software necessary to flash our firmware. You need to download and use the DfuSe USB device firmware upgrade (UM0412).

The STSW-STM32080 package contains all binaries and source code for DfuSe USB device firmware upgrade (DFU) software, including the demonstration, debugging GUIs and protocol layers. It includes the DFU driver compatible with the latest Microsoft®OS. The DfuSe utility can be used to interact with the STM32 system memory bootloader or any In-Application Programming (IAP) firmware, running from the user Flash, thus allowing internal memories programming through the USB.

All source files for Microsoft®Visual Studio 2012 are provided as well, to allow the customization of the default GUI interface.

The STSW-STM32080 package is a free software PC utility from STMicroelectronics, which runs on Microsoft Operating Systems and communicates through the computer's USB RS232 with the STM32 microcontroller system memory boot mode. This software utility also contains a command line version and it is provided with Microsoft Visual Studio 12 source code.

While the software is free, when you download it, you will need to accept a licensing agreement and provide a name and email address. Click on the Download button to download the STM32080 package:

- The STTube driver consisting of the following two files:
 - STTub30.sys: Driver to be loaded for demo board.
 - STDFU.inf: Configuration file for driver.
- DfuSe_Demo_V3.0_Setup.exe: Installation file which installs the DfuSe applications and source code on your computer.

When the software is successfully installed, click the "Finish" button. You can then explore the driver directory.

The driver files are located in the 'Driver' folder in your install path: C:\Program files\STMicroelectronics\DfuSe).

Source code for the Demo application and DfuSe library is located in: C:\Program Files\STMicroelectronics\DfuSe\Sources

Documentation is located in:

C:\Program Files\STMicroelectronics\DfuSe\Sources\Doc

Connect the Gotek to a spare USB port on your PC. If the host computer detects the Gotek, the computer will start the 'Found New Hardware Wizard'. Select 'Install from a list or specific location' and click [Next] to continue.

The next window asks to Search for the best driver or not. Select 'Don't search. I will choose the driver to install'. Click [Next].

If the driver is already installed, the next window shows the model list of compatible hardware models. The computer should show a "STM Device in DFU Mode" or similarly named device.

If this does not appear, click on [Have disk].

In the 'Install From Disk' dialog box, click [Browse...] to specify the driver files location; the driver directory is located in your install path: C:\Program files\STMicroelectronics\DfuSe\Driver

Then click [OK]. The PC autoselects the correct INF file, in this case STDFU.INF. Once Windows has found the required driver .INF file, the compatible hardware model will be displayed in the model list.

Click [Next] to proceed. You should see a warning message about not passing a Windows Logo testing, but click [Continue Anyway]. Windows should then display a message indicating that the installation was successful. Click [Finish] to complete the installation.

What is a DFU file?

Users who have purchased DFU devices require the ability to upgrade the firmware of these devices. Traditionally, firmware is stored in Hex, S19 or Binary files, but these formats do not contain the necessary information to perform the upgrade operation; they contain only the actual data of the program to be downloaded. However, the DFU operation requires more information, such as the product identifier,

vendor identifier, Firmware version and the Alternate setting number (Target ID) of the target to be used, this information makes the upgrade targeted and more secure. To add this information, a new file format was required, to be called the DFU file format.

We are now ready to actually reprogram the Gotek.

Open the DfuSe Demo (v3.0.5) program. A new window appears:

DfuSe Demo (v3.0.5)

Available DFU Devices

☒ *STM Device in DFU Mode*

☒ *Supports Upload*

☒ *Supports Download*

☒ *Can Detach*

☐ *Manifestation Tolerant*

☐ *Accelerated Upload (ST)*

☐ *Leave DFU Mode*

Actions

Select Targets(s)

Target Id	Name	Available Sectors...
<input checked="" type="checkbox"/> 00	Internal Flash	128 sectors
<input checked="" type="checkbox"/> 01	Option Bytes	1 sectors

Upload Action

File:

Upgrade or Verify Action

File (FileName).dfu

Vender ID [0483] Targets in file:

Product ID [0000] [00 ST]

Version [0000]

☐ *Verify after download*

☐ *Optimize Upgrade duration (Remove some Ffs)*

Transferred data size

0 KB(0 Bytes) of 0 KB(0 Bytes)

Operation duration

00:00:00

Click on [Choose] at the bottom of the window. In a new window, browse to find and click on your 'FF_Gotek-v3.23.dfu' file in the FlashFloppy-v3.23 folder, probably in the host computer's Documents folder. Now click on the [Upgrade] button of the previous window.

You will get a message window:

DfuSeDemo

Your device is read protected.

Would you remove the read protection?

Note: After this, you can **never** restore the original Gotek Firmware.

Click [Yes]. You will get a message window:

DfuSeDemo

Your device was plugged in DFU mode.

So it is impossible to make sure this file is correct for this device.

Continue, however?

Click [Yes] again. The bottom of the DfuSeDemo window shows the update's progress in a green bar for the "Target 00: Upgrading - Erase Phase (xx%)" . Then the same bar changes to "Target 00: Upgrading - Download Phase (xx%)" . Then we are done.

Future Firmware Updates:

If you have previously flashed the full firmware, you can make future updates with a USB Flash Drive. When new firmware updates are released, they are much easier to install than the process detailed above (i.e. using extra jumpers, computer programs, etc...)

When instructed to press both buttons, press Prev and Next. You may alternatively press Select on the rotary encoder, if you have it.

- Remove all old *.upd files from the Root directory of the USB flash drive.
- Copy the *.upd file from the root directory of the release archive on your PC computer to the Root of the USB flash drive.
- Insert the USB flash drive into the Gotek and power ON with both buttons pressed.
- You should now be in the bootloader (The Gotek display shows "uPd" or "FF Update Flash" in the OLED display).
- Release both buttons.
- The latest firmware will be programmed and the Gotek will reboot into it.
- The Gotek may lastly display E34 if no image files are found in the USB flash drive. This is normal. Hopefully, success!

Errors during an update are reported on the LED display:

- E01 No update file found
- E02 More than one update file found
- E03 Update file is invalid (bad signature or size)
- E04 Update file is corrupt (bad CRC)
- E05 Flash programming error
- E34 No image files were found in the present USB FDrive
- Fxx FatFS error (probably bad USB drive or filesystem)

Updating the Bootloader

The main bootloader can also be updated by USB flash drive, but new features and bug fixes are rare. In general, you do not need to update the bootloader and can skip these steps.

Note: It is important for this operation to succeed, otherwise the Gotek will require a full reflash by following the steps in Initial Programming. Therefore make sure the Gotek has a stable power supply and a known good USB flash drive.

- Remove all old *.upd files from the Root directory of the USB flash drive.
- Copy the *.upd file from the alt/bootloader/ subfolder of the release archive on your PC to the Root of the USB flash drive.

- Insert the USB flash drive into the Gotek and power ON with both buttons pressed.
- You should now be in the old bootloader (display shows "uPd" or "FF Update Flash"). Release both buttons.
- The latest bootloader will be programmed and the Gotek will reboot into it.
- The display should again show "uPd" or "FF Update Flash".
- Remove the USB flash drive and follow the steps for normal firmware update.

In the unlikely case that an error occurs during update, "ERR" will be displayed. At this point the Gotek will require a full reflash as the old bootloader has been erased.

FlashFloppy Firmware

Reference: github.com/keirf/FlashFloppy/wiki/Firmware-Update

As mentioned at the start, there are two viable firmware upgrades available that promise several improvements over the standard 1.44 Mb 3" drive emulator; the FlashFloppy Firmware upgrade and the HxC Firmware upgrade.

Both firmware upgrades treat the USB flash drive completely different from the Basic Gotek floppy disk emulator discussed in the last article. They were apparently developed by gamers for gamers, in that they used compressed disk images of floppy disks to be installed in the Root directory of the USB flash drive. From there, each disk image was automatically placed into separate accessible partitions in the Gotek's USB flash drive.

These compressed disk images had several advantages:

- They could be easily created using Disk Image Software.
- They were in a compressed format, but could be played as if the individual game files were on a floppy disk.
- They could be sold or given away as groups of games for the Atari, Amiga, and other gaming computers.

Note: These disk images made it possible to select a game (or a disk of files) by using the rotary encoder, and playing the game as though it were the original floppy disk. It required no further uncompressing nor manipulation.

Other advantages of the new Firmware were:

- Operation with a rotary encoder, an improvement over the two buttons of the basic Gotek emulator.
- Operation with a speaker for floppy disk sounds.
- Operation with an OLED display, displaying names of disks rather than just partition numbers.
- Operation with all common 8", 5" and 3" disk drive formats.
- Operation exactly as a floppy drive would, with files and folders.
- Flexibility, which was created through the use of additional files:
 - = FF.CFG, which controls the configuration of the firmware to use the upgrades mentioned, and
 - = IMG.CFG, which sets the disk size parameters.

Disadvantages? Well, at least in my mind:

- Setup has become very complicated.
- The websites of both firmware upgrades are complicated and completely lacking of any organization. If you are new to this Gotek, and gaming as a whole, expect a steep learning curve.

- Partitions are only created individually, created by the firmware automatically by the number of image files loaded in the USB flash drive's Root directory. Only have 10 images? All you get is 10 partitions. Want all 1000? Install 1000 blank disk images!
- For use as a normal disk drive, the Root directory must contain at least one blank disk image, which tells the firmware to create an empty disk partition for your use.
- Flexibility is created through the use of additional files, FF.CFG and IMG.CFG, which increases setup complexity.

So, what does all this mean?

Well, for someone like me with nearly 1000 floppy disks stored safely under the bed in the guest room, the basic Gotek sounds like the perfect answer - the disks already have disk numbers, and as infrequently as they are needed, even having to use two buttons to select the disk from a paper index is a piece of cake.

For someone who wants to play games from a compressed set of games listed on Ebay, or other gaming sites - the new firmware is ideal. But, we really do not have a whole lot of games usable on the Z-100, and to create disk images of each BASIC game routine would be a bit of overkill, and a lot of work for someone to create.

Which brings us to the average Z-100 user who just wants to eliminate the floppy disks on the shelf. You should upgrade the firmware IF:

- You need to use the Gotek to emulate disks other than 3".
- You would like an OLED display, instead of a number.
- You get tired of using the two buttons to navigate.

So, let's take a close look at our first firmware update, FlashFloppy, and I'll try to soften the learning curve. Best of all, it is free!

The latest download is: FlashFloppy-v3.23.zip.

FlashFloppy is compatible with any Gotek device with a 34-pin header, all of which share the SFR model prefix (eg. SFR1M44-U100K, SFRM72-U100; though not all sellers mention the model name explicitly). These 34-pin models all share the same basic design, varying only in pre-installed firmware and user interface (display and buttons). Since the fixed-format (720kB vs 1.44MB) stock firmware is being replaced, this particular distinction between Gotek models is irrelevant: all of them can handle an almost infinite range of disk formats with FlashFloppy installed. Unless performing display and rotary encoder upgrades, choose a model with three-digit LED display.

Programming the Firmware

The first installation of this Firmware to Gotek can be done either by serial or USB link to a host PC. These methods were discussed in detail in the last section.

Initial Setup:

Physical Interface

By default FlashFloppy will emulate the Shugart floppy interface for a 3", 135 tpi,, 1.44Mb high density drive. This is compatible with a broad range of systems including Amiga, Atari ST, Amstrad CPC, and many other devices. Shugart-compatible systems will typically expect the Gotek to respond as 'unit 0'. Therefore place the selection jumper at location S0 at the rear of the Gotek.

IBM PC compatibles use a slightly modified interface which places the disk-changed signal on a different pin. To select this interface mode place a jumper at location JC at the rear of the Gotek. The host system

may expect the Gotek to respond as 'unit 1': in this case place the selection jumper at location S1 at the rear of the Gotek.

For use in the Heath/Zenith Z-100 series computer, the Gotek can use either S0 or S1, leaving your real floppy drive as A: or B:.

An alternative method to specify the interface mode (Shugart or PC) is via the FF.CFG file, which we will discuss in a minute.

USB Storage

FlashFloppy requires a USB flash drive formatted with FAT32 (note that ExFAT and NTFS are not supported). Most USB flash drives are factory-formatted appropriately.

Configuration Files

FlashFloppy has a number of optional configuration files. We will only concentrate on those useful for the Z-100. As we already mentioned, if the drive you wish to use is the default 3" high density drive, we need no further configuration files. However, if you wish to emulate a different drive system, you will need one of more of these:

- FF.CFG: Global configuration of drive emulation, display handling, rotary encoder, sound, etc.
- IMG.CFG: Manual configuration of IMG/IMA ("Raw") track layouts for various drive sizes and formats.
- IMAGE_A.CFG / INIT_A.CFG: Last/initial image to load

Note: FF.CFG and IMG.CFG must be placed in the Root directory or the FF/ subdirectory of your USB flash drive via the USB port of your PC, before use in the Gotek Floppy Disk Emulator! Further, if FF/ exists, FF.CFG and IMG.CFG must reside there, as the Root folder is not searched. **As I see NO advantage of a FF/ subdirectory, let us agree to just use the Root Directory.**

Image Navigation Modes

FlashFloppy supports three different operating modes. For the best experience across the widest range of host systems, Native mode is recommended in combination with an LCD or OLED display.

- **Native mode:** No need for any configuration files. Allows direct selection of any valid image file on your USB flash drive.
- **Indexed mode:** Switches between image names of the form DSKA0000, DSKA0001, and so on.
- **HxC Compatibility mode:** A legacy mode requiring an image-selector program to run on the host system to pre-assign disk images to numbered slots or partitions on the USB flash drive.

Native mode

In this mode you need NO configuration files or selector software (however, as in all operating modes, you may optionally configure FlashFloppy via FF.CFG).

Behavior depends on which display type you have connected:

- **LCD/OLED:** FlashFloppy will allow you to select between any valid image file and any folder within the current directory. Selecting a folder will make that the new current directory, and list any image files and subfolders within.
- **7-Segment LED:** FlashFloppy will automatically assign all valid images in the Root folder of your USB flash drive to partitions which you can switch between using the Gotek buttons.

In all cases, the following folders and images are ignored and will not be listed for navigation:

- Any name beginning with a dot (.)
- Any file which is not a valid image (eg. unrecognized filename extension)
- Any folder or image with the FAT "Hidden" attribute set
- Root folder called FF/
- Any folder called __MACOSX/ (these get strewn around by macOS)

Indexed mode

This mode is configured within FF.CFG by the command:
nav-mode = indexed.

FlashFloppy will switch between image names of the form DSKA0000*.*, DSKA0001*.*, and so on. Each image is automatically assigned to the corresponding numbered partition on the USB flash drive in the Gotek. Any supported image type can be used.

Note: An arbitrary filename is allowed after the DSKAnnnn prefix (eg. DSKA0000_myimage.img). Additionally, the DSKA prefix can be changed within FF.CFG with the command: (indexed-prefix=)

HxC Compatibility mode

This mode selects between pre-assigned partitions in an HXCSDFE.CFG file. These slots are configured by an image-selector program running on the host system, either in ROM or (more typically) an AUTOBOOT .HFE image compatible with your system, copied to the Root directory of your USB flash drive. Supported systems include Amiga, Atari ST, and Amstrad CPC. The correct files for Amiga and Atari ST are included in the FlashFloppy release distribution.

The Gotek buttons cycle between the assigned partitions in the HXCSDFE.CFG file. To reassign partitions, boot the file selector; this is immediately accessible in partition 000 by pressing both Gotek buttons at any time. Configuration with the host selector software is straightforward; assign files from your USB flash drive to drive A "partitions". On reboot, these partitions are accessible via the up and down buttons of the Gotek. Holding a button will cycle faster through the populated partitions. Pressing both buttons will take you to the file selector.

Important Note: HxC assigns images to partitions by physical location on the USB flash drive, rather than by full path name. This means that if you move files around within the file system, the partition assignments will break with unpredictable results (FlashFloppy error messages, garbage disk images, etc). You must not copy a new image over an old, or file-by-file copy one USB flash drive to another, unless you subsequently refresh the partition assignments in HXCSDFE.CFG.

Note: In HxC Compatibility mode, the files HXCSDFE.CFG and AUTOBOOT.HFE must always be in the Root folder.

FF.CFG Configuration File

FlashFloppy has many configurable settings which can be specified in an INI-style configuration file called FF.CFG, placed in the Root folder or FF/ subfolder of your USB flash drive. An example is provided in the examples/ subfolder of the FlashFloppy distribution.

Resetting to "Factory Default" Configuration

Out of the box, FlashFloppy uses the default values described in the Options list. When these are modified by values in FF.CFG they are recorded into the Gotek's flash memory and become the new default values when the drive is next powered ON.

If you wish to return to "factory defaults", press the Prev and Next buttons (or Select, or the rotary encoder, if you have them) for three seconds with no USB flash drive inserted. The display will show RST or Reset Flash Configuration; factory defaults are restored when the buttons are released.

FF.CFG: Example FlashFloppy Configuration File

```
## FF.CFG: Example FlashFloppy Configuration File
# Place in the Root folder or FF/ subfolder of your
# USB drive. But, if FF/ exists, FF.CFG must reside
# there, not the Root folder.
# Note: Hash tags represent comments & the comment
# lines may be omitted. (I recommend leaving them.)
# Un-commented lines below are the configuration
# variables (in boldface) and are default settings.
# They cannot be overridden by settings in other
# configuration files.
```

##

DRIVE EMULATION

```
Floppy-drive interface mode (interface pins 2 & 34)
# jc: Specified by jumper JC
# (open: Shugart, closed: ibmpc)
# shugart: P2=DSKCHG, P34=RDY (Amiga, Atari, others)
# ibmpc: P2=unused, P34=DSKCHG (IBM PC interface)
# ibmpc-hdout: P2=HD_OUT, P34=DSKCHG
# (not generally needed: prefer 'ibmpc')
# jppc: P2=unused, P34=RDY (Japanese PC standard)
# jppc-hdout: P2=HD_OUT, P34=RDY
# (Japanese PC alternate: prefer 'jppc')
# akai-s950: Legacy alias of 'jppc-hdout',
# previously used for Akai S950
# amiga: P2=DSKCHG, P34=DRIVE_ID
# (not generally needed: prefer 'shugart')
interface = jc
```

```
# Host platform: Improves image-format detection
# for generic types such as IMG
# acorn: Acorn ADFS
# akai: Akai (S01, S20, S950), Korg, SC Prophet 3000
# casio: Casio (FZ-1)
# dec: DEC (RX33, RX50)
# ensoniq: Ensoniq (ASR, TS, etc)
# fluke: Fluke 9100
# gem: General Music (S2, S3)
# ibm-3174: IBM 3174 Establishment Controller
# memotech: Memotech
# msx: MSX
# nascom: Nascom
# pc98: NEC PC-98
# pc-dos: PC DOS Format
# (geometry determined from Bios Parameter Block)
# tandycoco: Tandy Color Computer (CoCo)
# ti99: TI-99/4A
# uknc: UKNC / DVK Soviet PDP-11
# unspecified: Common default geometries
# (including IBM PC)
host = unspecified
```

```
# 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
```

```
# Forcibly write-protect images, or respect the
# FAT read-only attribute?
# Values: yes | no
write-protect = no
```

```
# Filter glitches in the SIDE-select signal
# shorter than N microseconds
# Values: 0 <= N <= 255
side-select-glitch-filter = 0
```

```
# Rotational offset of disk after a track change
# instant: No rotation during track change
# realtime: Emulate rotation of disk while
# track is changing
# Values: instant | realtime
track-change = instant
```

```
# Rotational offset of disk after draining a
# write to Flash
# instant: No rotation
# realtime: Disk rotates in real time during drain
# eot: Disk rotates to (near) end of track
# Values: instant | realtime | eot
write-drain = instant
```

```
# Index pulses suppressed when RDATA and
# WDATA inactive?
# Values: yes | no
index-suppression = yes
```

```
# Milliseconds from head-step start to RDATA active.
# Values: 0 <= N <= 255
head-settle-ms = 12
```

```
# Milliseconds delay from motor-on to drive ready.
# On a standard unmodified Gotek the motor signal
# is not connected and a non-default value here
# will have no effect. Most systems and software
# do not care about correct motor behavior, and
# default (ignore) works fine.
# Values: ignore | 0 <= N <= 1000
motor-delay = ignore
```

```
# What causes the disk-change (chg) signal to reset?
# step: Step command received
# pal4: CHGRST (pin 1 on old Sony drives),
# connected to PA14 (JTCK/SWCLK)
# delay-N: Automatic after N*0.5sec (0 <= N <= 15)
chgrst = step
```

##

STARTUP / INITIALIZATION

```
# Disk image loaded or ejected at startup?
# Values: yes | no
ejected-on-startup = no
```

```
# Which image (or folder) is selected at startup?
# last: Last-selected item at power-off
# (recorded in IMAGE_A.CFG)
# static: Static path specified in INIT_A.CFG
# init: First item in root folder
# Values: last | static | init
image-on-startup = last
```

```
# Time in milliseconds to attempt to
# probe attached display.
# You may set to 0 if you use a 2-digit LED display.
# Values: 0 <= N <= 65535
display-probe-ms = 3000
```

##

IMAGE NAVIGATION

```
# Auto-select the current file after N seconds
# N=0: disable auto-select
# Values: 0 <= N <= 255
autoselect-file-secs = 2
```

```

# Auto-select the current folder after N seconds
# N=0: disable auto-select
# Values: 0 <= N <= 255
autoselect-folder-secs = 2

# Sorting of folder entries; native navigation mode.
# always: Always sort folder entries.
# Large folders may be truncated.
# never: Never sort folder entries, use FAT order.
# small: Only sort folders which are small
# enough to sort in full.
# Values: always | never | small
folder-sort = never

# Priority of files vs subfolders when sorting:
# folders: Folders listed before files
# files: Files listed before folders
# none: Files and folders are not differentiated
# Values: folders | files | none
sort-priority = folders

# 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

# When navigating slots or folder, loop at min/max?
# Values: yes | no
nav-loop = yes

# 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

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

```

```

##
## DISPLAY
# Display Type.
# auto:      Auto-detect (7-seg LED, LCD, OLED)
# lcd-CCxRR: CCxRR backlit LCD with I2C
#             backpack (16<=CC<=40, 02<=RR<=04)
# oled-128xNN: 128xNN I2C OLED (NN = 32 | 64)
# -rotate:    OLED view is rotated 180 degrees
# -narrow[er]: OLED view is restricted to
#             Gotek display cutout
#             (-narrow: 18 chars; -narrower: 16 chars)
# -inverse:   Inverse/reverse video
#             (black text on white background)
# -ztech:     ZHONGJY TECH 2.23" 128x32
#             SSD1305 OLED display
# Values:     auto | lcd-CCxRR | oled-128xNN
#             [-rotate][-narrow[er]]...
display-type = auto

# 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 = 6x13

# OLED contrast/brightness.
# Values: 0 <= N <= 255
oled-contrast = 143

# Text height and arrangement on LCD/OLED
# 'default', or a comma-separated list
# (one entry per LCD/OLED row, top down).
# Each list item is a digit plus optional
# height specifier: <content-row>[d]
# content-row: '0-3' = specified content row,
#              '7' = blank
# 0: Current image name
# 1: Status
# 2: Image/Volume info
# 3: Current subfolder name
# height specifier: 'd' = double height
# (32px, OLED only; ignored for LCD)
# 'default' depends on display,
# eg.: oled-128x32='0,1' ; oled-128x64='3,0d,1'
# Values: [0-7][d] | default
display-order = default
# Note: Entering 2 here caused my OLED to display
# "Raw DS/DD 84T" & "Free:3.9/4.0G"

# 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 = 60

# Switch ON LCD/OLED display if drive activity?
# yes: Trigger on track changes and disk writes
# sel: Trigger on drive select
# no: No automatic trigger
# Values: yes | sel | no
display-on-activity = yes

# LCD/OLED long filename scroll rate in ms/update
# Values: 100 <= N <= 65535
display-scroll-rate = 200

# LCD/OLED pause time (msec) at start/end of scroll.
# Zero means endless scroll
# Values: 0 <= N <= 65535
display-scroll-pause = 2000

# LCD/OLED long filename scroll rate
# during navigation (ms per update)
# Values: 0 <= N <= 65535
nav-scroll-rate = 80

```

```
# LCD/OLED long filename pause before scroll,
# during navigation (milliseconds)
# Values: 0 <= N <= 65535
nav-scroll-pause = 300

##
## MISCELLANEOUS
# Speaker volume for head STEP
# Values: 0 <= N <= 20
step-volume = 10

# Report specified version number to host software
# Values: <quoted-string>
# ("" means report real version)
# eg. da-report-version = "v3.0.0.0"
da-report-version = ""

# Automatically extend certain types
# of truncated image file (SSD,DSD,TRD)?
# Values: yes | no
extend-image = yes
```

IMG.CFG Configuration File

FlashFloppy can usually automatically determine the layout of IMG/IMA/DSK files, based on their size and (optionally) a hint about the host they target (FF.CFG: host=).

However, there are such a wide range of floppy drive sizes and systems that FlashFloppy cannot hope to automatically support them all. For example, the CP/M operating system does not have a standard disk layout, and these systems, among others, require a degree of manual configuration.

The solution is an INI-style configuration file called IMG.CFG, placed in the Root folder or FF/ subfolder of your USB flash drive. If FF/ exists, IMG.CFG must reside there.

General Notes

Most systems require a Shugart interface (the default) and drive select 0 (jumper on S0). Such systems will work out of the box with a factory-fresh Gotek programmed with FlashFloppy firmware, requiring only a physical jumper at the rear to be moved from S1 to S0.

If this does not work then also try the following:

- Jumper at S1 only
- Jumpers at JC and S0
- Jumpers at JC and S1

If the system still does not work and the Gotek green LED remains lit continuously, try connecting the ribbon cable 'upside down'.

Note: In most cases FlashFloppy requires a straight floppy cable with no 'twist' between the connectors. If you do use a twisted cable, you should jumper MO instead of S0. The exception is the one case that the twist was designed for - connecting FlashFloppy as Drive B to a PC-compatible floppy header.

Note: For use in the Z-100, if you program a separate (recommended) USB flash drive for a specific drive size and/or type, such as 8", then the IMG.CFG file will have parameters adjusted for that particular drive. Do NOT mix them up. I suggest placing each USB flash drive's Root directory files in the Documents Folder of your host computer under it's own name. For example, I have a folder 'GotekUSBFlash Drives' containing the separate folders:

```
Gotek3Drive1440 (for HD 3", 135 tpi, 1.44Mb drive)
Gotek5Drive1200 (for DD 5", 96 tpi, 1.2Mb drive)
Gotek8Drive1200 (for DD 8", 48 tpi, 1.2Mb drive)
```

Note: I have added additional lines in *italics* to show the drive options in IMG.CFG that you must pay attention to. In each case, select **ONLY ONE** option in your IMG.CFG file. While you may delete all lines beginning with the # sign, I recommend leaving the entire IMG.CFG file intact, to make understanding the values easier in the future.

Note: This IMG.CFG file is optional for the HD 3", 1.45Mb drive, which is the default configuration for the Firmware upgrade.

IMG.CFG: Example FlashFloppy Configuration File

```
## IMG.CFG: Example FlashFloppy Geometry Configuration
## File for Raw IMG Images.
## I have included several lines to help you create
## your own specific IMG.CFG file. Choose or modify
## one of the following to describe your drive type:
## Set for 3", 135 tpi, HD, 300rpm, 1.44Mb drive.
## Set for 5", 96 tpi, DD, 360rpm, 1.2Mb drive.
## Set for 8", 48 tpi, DD, 360 rpm, 1.2Mb drive.
# Supports tagged IMG/IMA raw image files
# with self-identifying geometry.
# If you wish to use this feature, your IMG.CFG must
# be placed in the Root folder or FF/ subfolder of
# your USB drive.
# Note: If FF/ exists, IMG.CFG must reside there,
# not the Root folder.
# A badly-defined section (eg. undefined mandatory
# parameters) may result in error 31 (Bad Image).
#####
## TAG in square brackets.
## Format: [<tagname>][:<filesize>]
## Matching Rules:
## 1. <tagname> matches images of the form:
##    *.<tagname>.{img,ima,dsk}
## 2. Missing (empty) tagname matches any
##    *.{img,ima,dsk} which is untagged
##    or does not match any other defined tag.
## 3. Tagnames and their matches are NOT case
##    sensitive.
## 4. <filesize>, if specified, must exactly match
##    the image filesize.
## 5. If an image matches no tag, FlashFloppy uses
##    normal geometry auto-detection based on the
##    host= setting in FF.CFG.
# These lines change considerably depending upon
# the drive size being emulated. Choose one:
# For 3", 1.44Mb drives:
# [dsdd80::1474560] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 1440kB (1474560 bytes).
[dsdd80::1474560]
# For 5", 1.2Mb drives:
# [dsdd80::1213952] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 1185.5kB (1213952 bytes).
[dsdd80::1213952]
# For 8", 1.2Mb drives:
# [dsdd80::1261568] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 1232kB (1261568 bytes).
[dsdd80::1261568]

## DISK-SCOPE PARAMETERS
# Apply to the whole disk image.
# Cannot appear in a @tracks sub-section.
# Mandatory: Number of cylinders (1-255).
# For 3", 1.44Mb drive, set number of cylinders as:
cyls = 80
# For 5", 1.2Mb drive, set number of cylinders as:
cyls = 80
# For 8", 1.2Mb drive, set number of cylinders as:
cyls = 77

# Mandatory: Number of heads (1-2).
heads = 2
```

```
# Image file track layout. Default is "interleaved".
# Comma-separated values:
#   sequential:      Sequential cylinder ordering:
#                   all side 0, then side 1.
#   interleaved:     Interleaved cylinder ordering:
#                   c0s0, c0s1, c1s0, c1s1, ...
#   reverse-sideN:   Side-N cylinders are ordered
#                   from high to low (N=0,1).
#   sides-swapped:   Sides 0 and 1 ordering is
#                   swapped in the image file.
#                   eg. "sequential,reverse-side1"
# file-layout = interleaved

## TRACK-SCOPE PARAMETERS
# These vary across tracks. By default, they apply to
# all tracks unless preceded by a @tracks declaration
# tracks = <track-list>
# Specify tracks that track-scope parameters apply to
# Format:
#   track-list      ::= <track-range>[,<track-list>]
#   track-range     ::= <cylinder>[.<head>] |
#                   <cylinder>-<cylinder>[.<head>]
#   cylinder        ::= [0-9][0-9]*
#   head            ::= 0|1
# If no head is specified, all heads are assumed.
# Examples:
#   "0-23.1,45.0" specifies head 1 of cyls 0-23 and
#                   head 0 of cyl 45.
#   "27-33" specifies all heads of cyls 27-33.
# Number of sectors per track (0-256). Default is 0.
# For HD 3", 1.44Mb drives set Sectors/Track as:
secs = 9
# For HD 5", 1.2Mb drives set Sectors/Track as:
secs = 15
# For HD 8", 1.2Mb drives set Sectors/Track as:
secs = 18

# Bytes/sector (128, 256, 512, 1024, 2048, 4096, 8192)
# Mandatory if @secs is non-zero.
# For HD 3", 1.44Mb drives set Bytes/Sector as:
bps = 512
# For HD 5", 1.2Mb drives set Bytes/Sector as:
bps = 512
# For HD 8", 1.2Mb drives set Bytes/Sector as:
bps = 1024

# ID of first sector on each track (0-255).
# Default is 1.
# Numbers may be expressed in hexadecimal with
# 0x prefix (eg. 0xab).
# id = 1

# Head number for all sectors on a track (auto|0|1)
# Default is auto.
#   auto = use physical drive head number
# h = auto

# Recording mode (fm | mfm). Default is mfm.
# mode = mfm

# Sector interleave. Default is 1:1 (no interleave).
# interleave = 1

# Sector skew per cylinders. Default is 0 (no skew).
# cskew = 0

# Sector skew per head. Default is 0 (no skew).
# hskew = 0

# Rotational RPM. Default is 300.
# For 3", 1.44Mb drive, set RPM as:
rpm = 300
# For 5", 1.2Mb drive, set RPM as:
rpm = 360
# For 8", 1.2Mb drive, set RPM as:
rpm = 360

# Post-Data Gap (auto|0-255). Default is auto.
#   auto = based on recording mode and sector size.
# gap3 = auto
```

```
# Post-Index Gap (auto|0-255). Default is auto.
#   auto = based on recording mode and sector size.
# gap4a = auto

# Index Address Mark (yes | no). Default is yes.
# iam = yes

# Data rate in kHz (kbit/s) (eg. MFM DD = 250).
# Default is 0.
#   0 = based on recording mode and size of track.
# rate = 0
```

Hardware Preparation

Format USB Flash Drive:

Note: If you had used the built-in self-format function to format USB flash drive into **1000** Partitions, each representing a 3.5" 1.44Mb Floppy Drive, the USB flash drive should be reformatted in the PC before use.

- Plug the USB flash drive into any USB port of a PC or clone. Depending upon the Operating System, the USB port will be detected as a USB flash drive. I'm running Windows 10.
- Right-click on the USB flash drive (or whatever name it may be) and select Format. The Format window will appear showing:

```
Capacity:           (Determined by size of USB flash drive)
File System:   FAT32   (default)
Allocation unit size   (4096 bytes on mine)
[ Restore device defaults ]
Volume label         (Whatever name you wish)
Format options
[X] Quick Format
```

[Start] [Close]

- Click on [Start] and the Formatting should be completed.

How we will finish the USB flash drive will depend upon your choice of floppy drive system to be emulated. We will discuss all the options for installation in the Heath/Zenith Z-100 series computer, beginning with FlashFloppy's the default 3" 1.44Mb high density drives, next.

Use in the Z-100 Series Computer

Preparation of the USB Flash Drive

Before use in the Gotek, each USB flash drive must be prepared by adding disk image files and possibly some configuration files to the Root directory of the device. This is easily done by inserting the USB flash drive in any USB port of a host computer, usually a PC, and copying a set of files into the device's Root directory. These files change for each drive size and configure the Firmware for the drive system to be emulated.

Note: If the USB flash drive had been used in the earlier basic Gotek, it had been divided into as many as 1000 partitions. This USB flash drive can be formatted back into the original large FAT partition by inserting it in the USB port of a PC or other host computer, and reformatting.

Note: The following configurations also assume that the only change to the Gotek has been to the Firmware. There is a separate article on Gotek Hardware Changes. My Gotek still has the original 3-digit, 7-

segment display and navigation buttons only. If you have already installed the upgrades to an OLED display, speaker, and/or rotary encoder, your results to certain operations may appear to be different from mine.

Note: The discussions here will concentrate on the use of these drives under the latest ZDOS v4 software and latest monitor ROM. Earlier ZDOS requires the use of setting the Z-207 Floppy Controller for 48 or 96 tpi 5" drives, using the DiskPack software floppy driver for drives not easily controlled by the BIOS, various cable configurations, and other notes or changes which were extensively addressed in the article, *"Gotek Floppy Emulator User's Manual"*, published in issue #133 of the *"Z-100 LifeLine"*. Please refer to that article for further information on use with these earlier versions of ZDOS.

High Density 3" 1.44Mb Emulation

As the 3" high density drive system is the default for the Gotek Floppy Disk Emulator, the only files necessary for successful use is placing at least one 3" blank disk image file in the Root directory of the USB flash drive.

These 3" blank disk images were provided in the download from HxC, **QuickInstall_FloppyDiskImages.zip**. Unzip this file and select the images from the folder "SDHxCFE_PC_1.44MB.ZIP". There are 21, DSKA0000.HFE through DSKA0020.HFE. Copy them to the Root directory of the USB flash drive.

For each disk image file placed in the Root directory, the Firmware will create a partition accessible as a single 3" floppy disk. This floppy disk will act, in all regards, as a real floppy disk. It can be formatted, have files and directories, and be erased. The images must be placed in the Root directory of a USB flash drive in the USB port of a host computer, such as my Windows 10 PC, before moving the flash drive to the Gotek Emulator.

Note: For those using hardware upgrades to their Gotek, parameters can be adjusted if you also add a FF.CFG file in the Root directory.

The Z-100 series computer must also be prepared.

Using ZDOS v4 in the Z-100, DRIVECFG.COM could be configured to any drive letter, but I made my Gotek (as unit 1) configured as:

B: HGH Den. 135tpi 3 ½" 34 Z207p 1 AND
D: HGH Den. 135tpi 3 ½" 50 Z207p 3

This allowed me to use the Gotek on the Z-207 Floppy Controller's 34-pin interface or the 50-pin interface (with an adapter cable).

Using ZDOS v3 in the Z-100, the drive would require the use of the DISKPACK software driver with a similar cable configuration. For a complete discussion on the use of the DiskPack Floppy Disk Driver, please see this issue's insert, *"Z-100 DiskPack User's Manual"*.

According to the Diskpack Manual, the drive must be placed on the Z-207 Floppy Controller's 50-pin interface, under DiskPack control as unit 2 or 3 (drive letters begin with I:). The drive configuration is:

Sec	Sec/	Trk/	No.	Sec/	Sec/	Dir	FAT	Media	Std
Size	Trk	Side	Head	Cls	FAT	Ent	ID	Size	
512	18	80	2	2	9	224	F0	1440Kb	IBM (default)

Configure the drive in DISKPACK's DPCNFG utility as:

				UnFMT	Step	REC	Data
Size	Hds	TPI	Cap.	Rate:	Mode:	Trks:	
3.5"	2	135	2.0Mb	3 ms	MFM	80	

For those with the latest Z-DOS v4 and upgraded ZROM, and using a Z-207 Floppy Controller with the Barfield Modification, you simply plug the Gotek Floppy Disk Emulator into the 34-pin interface, specify a 3" High Density 135 tpi drive on the 34-pin connector in DRIVECFG and you should be good to go.

If you have no drive positions left on the 34-pin interface, or if you would prefer to use the 50-pin interface, specify the 3" drive on the 50-pin interface in DRIVECFG and you are ALMOST ready to go.

You will need to construct a 50-pin Z-207 Interface double row female connector to 34-pin double row female drive connector (standard 3" drive type). Depending upon the Gotek model, you may need to include a diode to simulate a drive Ready signal. The 34-pin connector does not use a Ready signal, but the 50-pin interface requires one. Please see the article, *"Gotek Floppy Emulator User's Manual"*, published in issue #133 of the *"Z-100 LifeLine"* for more information on this adapter cable and use.

Note: If you include the FF.CFG file in the Root Directory of the USB Flash Drive, you are able to **change the Pin 34 setting to RDY:**

```
# 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
```

Once placed in the Z-100, when DIR or ZDIR is run on each partition number, a volume name will be displayed and there will be a single README.TXT file (which are the same for all partitions). If you TYPE that file, a small text file From HxC will display.

The label can be changed with the LABEL command and the TXT file may be deleted. The Gotek partition can be treated as any 3" floppy drive. Our DIAGNOSTICS program works on the drive and even the RPM of the disk, 300 RPM, is reported!

Change the disk in the Gotek by simply pressing one or both of the navigation buttons; right increments the ones digit, left increments the tens digit, and pressing both will increment the hundreds digit.

CAUTION: To prevent possible damage to the Gotek, the USB flash drive or both, **DO NOT REMOVE the USB device** while power is applied to the Gotek! Shut the computer down.

CAUTION: To prevent loss of data or corrupting files, **DO NOT CHANGE the partition number** while the Gotek drive LED is lit! It should go out shortly.

High Density 5", 96 tpi, 1.2Mb Emulation

The 5" 1.2Mb high density drive system, as with all the other drive systems, require the special configuration file, IMG.CFG and possibly the FF.CFG file, in addition to their own, unique 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 5", 96 tpi, HighDen, 360rpm 1186Kb drive.
```

And change the IMG.CFG file to reflect the following:

```
# [dsdd80::1213952] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 1185.5kB (1213952 bytes).
[dsdd80::1213952]

# Number of sectors per track (0-256). Default is 0.
secs = 15

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

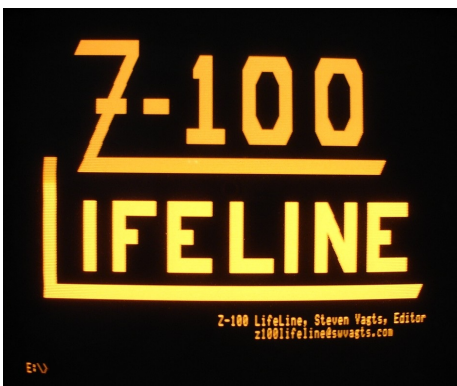
Note: For those using hardware upgrades to their Gotek, parameters can be adjusted in the FF.CFG file in the Root directory.

These 5" blank disk images were not provided in the download from HxC, or at least not that I could find. So I had to create my own. So, now is a good time to discuss creating our own blank disk images.

Creating Blank Disk Images

First, let me add that as editor of the "Z-100 LifeLine" I felt that I should use my own text file. So, dusting off my Paint100 Program, I created a text screen displaying Z-100 several characters high and LIFE LINE below it, also several characters high, then listed my name and other information in the lower right corner. It came out pretty well, if I do say so myself.

This picture was on a yellow monitor. I did not think anything of it until I was done with all the images, so I left it monochrome. Let me know if you would rather have a color screen.



I put my new screen, Z100LL.TXT, in the unmodified Gotek and copied it to my PC, but as it turns out, this will not work. We will see why in a minute...

Next, I double-clicked on the following series of folders to finally open the "HxCFloppyEmulator.exe" program:

Downloads > HxCFloppyEmulator_soft > HxCFloppyEmulator_Software > Windows > HxCFloppyEmulator.exe

This is an interesting program that allows us to manipulate or create disk images of files in a Floppy Disk. My PC doesn't have its own floppy disk, but it does recognize any partition in the Gotek as floppy drive A:, as it should. Anyway, here is the HxC control panel:

HxCFloppyEmulator v2.5.6.6"

<i>Floppy image</i>	<i>Settings</i>	<i>Look</i>	<i>Log</i>	<i>About</i>
[Load]			<i>Load a floppy file image</i>	
[Load Raw image]			<i>Load a custom raw floppy image/ Create a custom floppy</i>	
[Batch converter]			<i>Convert multiple floppy file image</i>	
[Disk Browser]			<i>Create/Browse a DOS floppy disk</i>	
[Export]			<i>Export/save the loaded file Image</i>	
[SD HxC Floppy]			<i>Configure the SD HxC Floppy Emulator</i>	
[Emulator settings]				
[USB HxC Floppy]			<i>Configure the USB HxC Floppy Emulator</i>	
[Emulator settings]				
[Floppy disk dump]			<i>Read a real disk</i>	
[Track Analyzer]			<i>Low level tracks viewer</i>	

No disk loaded.

Clicking on [Disk Browser] brings up another window:

DOS Floppy Disk File Browser

File System Type:

[3"5 720KB DSDD FAT12"] X [Create Disk]
[] [Get Files]
[] [Delete]
[] To add your files
[] to the disk just
[] Drag&Drop them
[] on the file browser
[] to the left!
[] [Load Image]
[] [Save/Export]
[No DOS image Loaded !] [Close]

Here's the neat part. Following the 'File System Type:', to the right, is a pair of up/down arrows. If you click on this, you can select from 39 disk types, and load a blank image! Select the desired disk and the File System Type is updated with:

[5"25, 1.2Mb DSHD FAT12"]

In the open window, I dragged my file name "Z100LL.TXT" from my floppy drive A: (it would not move it from my hard drive because it was not a floppy?) into the window and press the [Create] button and it was done!

The last line of the window updates:

```
[ Disk size: 1.15MB | File size: 34B | Left space: 1.12 | ]
```

Press [Save/Export] and you can save the file with any name in the PC's folder or the USB drive. Anticipating having sets of images for each type of drive, I saved it in my hard drive folder:

```
C:\Documents\GotekUSBFlashDrives\Gotek5Drive1200
```

Copy the file as many times as needed to create the additional blank disks. I used the file names: ZFD12_0000.hfe thru 0030.

Finally, I copied the set of image files to the Root directory of my USB flash drive, with the appropriately modified FF.CFG and IMG.CFG files, and I was done! Pretty neat.

It was so easy, I went back and changed the first set of disks and created the files needed for the other sizes I may use.

I placed the configured USB in the Gotek, and now had my 5" disk emulator ready for use on the Z-100.

Using ZDOS v4 in the Z-100, DRIVECFG.COM could be configured to any drive letter, but I made my Gotek (as unit 1) configured as:

```
B: HGH Den. 96tpi 5.25" 34 Z207p 1      AND
D: HGH Den. 96tpi 5.25" 50 Z207p 3
```

This allowed me to use the Gotek on the Z-207 Floppy Controller's 34-pin interface or the 50-pin interface, with an adapter cable.

Using ZDOS v3 in the Z-100, the drive would require the use of the DISKPACK software driver with a similar cable configuration. For a complete discussion on the use of the DiskPack Floppy Disk Driver, please see this issue's insert, "**Z-100 DiskPack User's Manual**".

According to the Diskpack Manual, the drive must be placed on the Z-207 Floppy Controller's 50-pin interface, under DiskPack control as unit 2 or 3 (drive letters begin with I:). The drive configuration is:

Sec	Sec/	Trk/	No.	Sec/	Sec/	Dir	FAT	Media	Std
Size	Trk	Side	Head	Cls	FAT	Ent	ID	Size	
512	15	80	2	1	7	224	F9	1200Kb	IBM (default)

Configure the drive in DISKPACK's DPCNFG utility as:

	UnFMT	Step	REC	Data		
Size	Hds	TPI	Cap.	Rate:	Mode:	Trks:
5.25" 2	96	1.6Mb	3 ms	MFM	80	

Operation in the Z-100 is similar to that described for 3" emulation.

Low Density 5", 96 tpi, 720Kb Emulation

The 5" 720Kb low density drive system, as with all the other drive systems, require the special configuration file, IMG.CFG and possibly the FF.CFG file, 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 5", 96 tpi, LowDen, 300rpm 720Kb drive.
```

And change the IMG.CFG file to reflect the following:

```
# [dsdd80::729088] matches images with names of
# the form: *.dsdd80.{img,ima,dsk}
# and with size exactly 720kB (729088 bytes).
[dsdd80::729088]

# Number of sectors per track (0-256). Default is 0.
secs = 19

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

Note: For those using hardware upgrades to their Gotek, parameters can be adjusted if you also add the FF.CFG file in the Root directory.

While these 5" blank disk images were provided in the download from HxC, I created my own, as discussed above. These were saved in my hard drive folder:

```
C:\Documents\GotekUSBFlashDrives\Gotek5Drive720
```

Copy the file as many times as needed to create the additional blank disks. I used the file names: ZFD720_0000.hfe thru 0030. Then copy the set of image files to the Root directory of the USB flash drive, with the appropriately modified FF.CFG and IMG.CFG files.

I placed the configured USB in the Gotek, and now had my 5" disk emulator ready for use on the Z-100.

Using ZDOSv4 in the Z-100, DRIVECFG.COM could be configured to any drive letter, but I left my 5" 360Kb drive as unit 0 and made my Gotek (as unit 1) configured as:

```
B: LOW Den. 96tpi 5.25" 34 Z207p 1
```

Using ZDOSv3 in the Z-100, the drive may be used on the 34-pin connector, **IF** the DIP switch on the Z-207 Floppy Controller is set for 96 TPI operation. Any real 5" floppy drive would also need to be a 96 TPI drive, using High Density diskettes.

Operation in the Z-100 is similar to that for normal real drives.

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
```

Emulating Other Sizes

The Gotek may be used to emulate other formats and smaller sized drives. Like I mentioned, there were 39 options in the HxC Emulator software. But I see little sense in using smaller capacity drives on the H/Z-100 series computers. Nevertheless, the procedures are the same if you have a need for a replacement 360Kb floppy drive, or need to maintain compatibility with another brand of computer.

Closing:

This is still NOT the end of the story. I just ran out of room and saved the best revelations for the next article, "Gotek Floppy Emulator Hardware Upgrades", where we will finally reveal it ALL. You will NOT want to miss it.

Note: You do not need to create a disk image, to back up a disk.

So, I have not addressed the procedures used to create disk images of your old software, as it is outside the scope of my articles. 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

