GPIB Flash Drive User Guide for Tektronix 4050 computers

ABSTRACT
User Guide for my GPIB Flash Drive designed to support Tektronix 4051, 4052/4052A, and 4054/4054A computers.

Monty McGraw

June 7, 2022

Table of Contents

1. Background

Tektronix 4051 computer system was introduced in 1975 at the dawn of the microcomputer. The 4051 was a complete system including a Motorola 6800 microprocessor, up to 32KB of RAM, 32KB of BASIC ROM, integrated 1024x800 vector graphics with 11-inch direct-view storage tube (DVST) monitor, internal tape drive using 3M DC300 300KB data cartridges, integrated keyboard, graphics hard copy interface, joystick interface, GPIB interface, option ROM backpack with two-slots, and optional RS-232 serial interface.

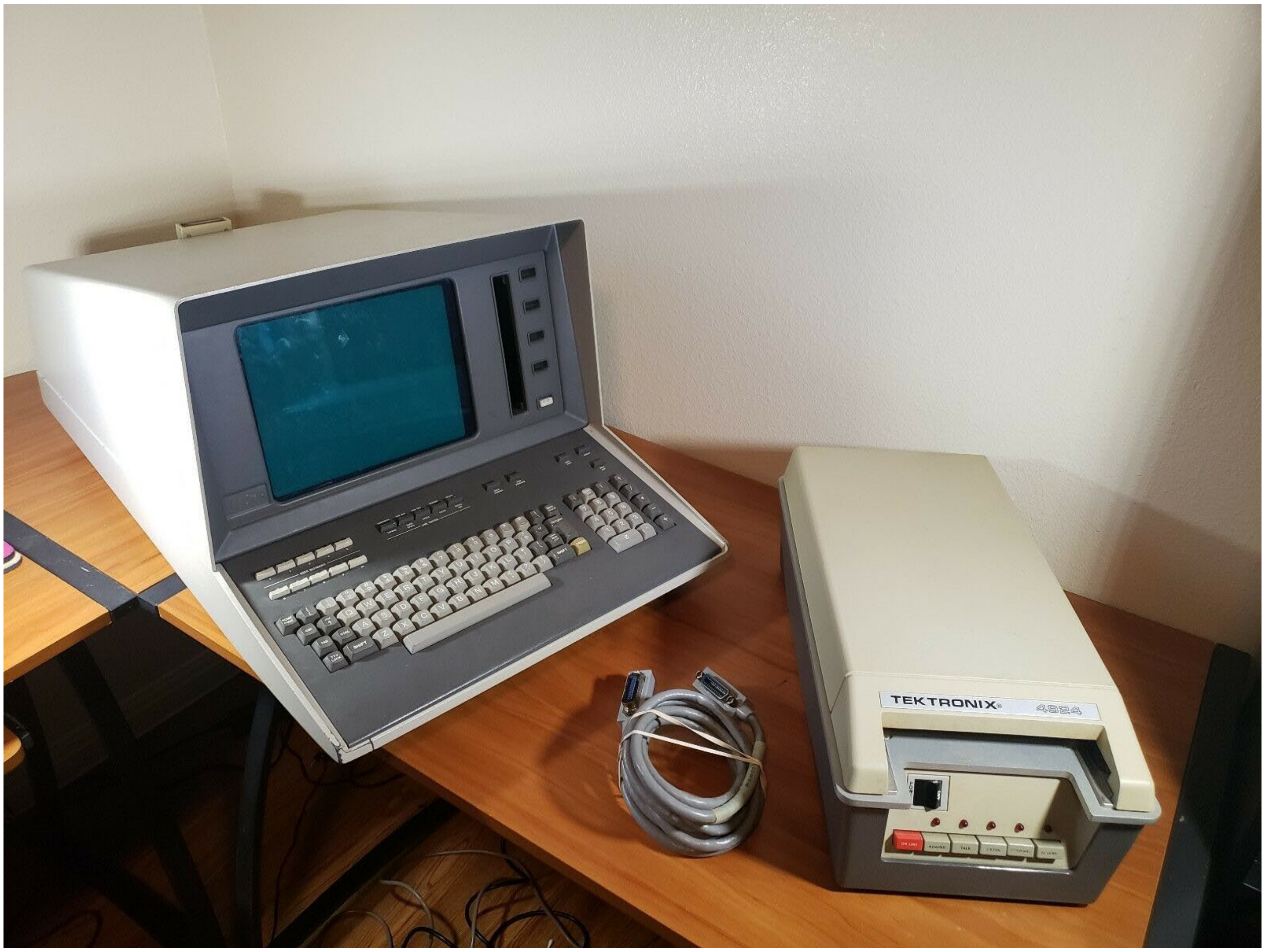

Figure 1 - Tektronix 4051 computer with optional Tektronix 4924 GPIB tape drive and GPIB cable

I used a Tektronix 4051 for three years at work in the late 1970's and developed several graphics simulation programs including submission of a Cubic Spline Interpolation program to the Tektronix 4050 Application Library. This was the first computer I had ever used directly and can be considered one of the first graphics workstation computers.

I collected and repaired a Tektronix 4052 computer in 2000, followed by a Tektronix 4054 computer, and found two other people that had collected 4051 computers. We began trying to share 4051 tapes we each had with various 4051 games and discovered that duplicating DC300 tapes was the only reliable way to share. As a computer design EE at Compaq Computer, I began trying to use the integrated GPIB interface connected to my PC using the parallel port but quickly found the lack of PC support for GPIB made the GPIB interface too difficult to use.

I then wrote a BASIC program for my Tektronix computer to send files from Tektronix tape over RS-232 to my PC and found that the Tektronix made heavy use of control characters for printing – which I encoded with my BASIC program into a 3-character sequence that I believed would not occur in Tektronix programs: "~X~" where X was the ASCII character typed on the Tektronix with the CTRL key. This serial program was cumbersome to use on more than a couple of files on a tape, so I did not recover many programs to my PC using this technique.

After copying several tapes and sending them to my new 'Tektronix' friends, I lost interest in my Tektronix computers until I discovered and joined the vcfed.org website in 2018 and found a new set of people that had collected and restored Tektronix 4050 computers and had attempted to archive a couple of data tapes to their PCs.

I then dusted off my Tektronix 4052 and 4054 and found they both needed some repairs – but worse that that I found my data tapes that had worked in 2000 had broken tape belts and could not be used. Researching this issue on the web, I found this was common, and although other materials had been tried as data cartridge tape belt replacements – they had mixed results including short life.

During this search I encountered several websites that offered solid state drives as replacement for these tape drives, however it appeared these drives only worked with specific computers – such as vintage HP or Commodore computers, and I could not find anyone that had developed a solution for the Tektronix 4050 series computers.

I posted on vcfed.org a thread "Tektronix 405x GPIB Flash Drive" in June 2018 and got several strong positive reactions, and in my second post in this thread I referenced a website that had used an Arduino to create a GPIB controller. I wired up an Arduino based on their instructions – but the Tektronix 4050 computers could only communicate with GPIB devices – NOT other controllers.

I continued searching for GPIB "device" programs, but never found any that could be used.

In the meantime, I had purchased a couple of NOS DC6250 Tapes in boxes and found their drive belts were intact and I could transplant the belts into my older DC300 cartridges and read those old tapes again! I began manually recovering individual files on each tape and posting them to a new Tektronix 4050 program repository I created on github:

https://github.com/mmcgraw74/Tektronix-4051-4052-4054-Program-Files

I also collected a Tektronix 4924 Tape Drive – which came with the service and user manuals! This tape drive was supported by Tek 4050 BASIC, and I had found tape copy programs that used the internal tape and external 4924 tape drive to make tape copies very easy. The 4924 service manual included an appendix showing each of the supported GPIB commands in "4051 mode" which was very helpful in developing the GPIB Flash Drive code.

I also collected a couple of Tektronix 4907 8-inch floppy drive systems, got them working with my 4054 and considered having the GPIB Flash Drive be compatible with the 4907 which provided a hierarchical file system. However, as I used the 4907, I realized it would NOT be a good fit for the GPIB Flash Drive project, as it not only required the File Manager ROM Pack, it was NOT compatible with any of the software programs that ran from tape without adding several 4907 commands to every program. So I decided to stick with my original plan to keep the compatibility with the internal and external (4924) tape commands.

When I discovered the AR488 GPIB project https://github.com/Twilight-Logic/AR488 in 2020 - and the associated thread on eevblog: https://www.eevblog.com/forum/projects/ar488-arduino-based-gpib-adapter/ I joined the eevblog thread and built an AR488 adapter with a ProMicro and Oshpark PCB. I was successful in getting the AR488 ProMicro to communicate with an HP meter as the controller – but it still did not have any ability to be a device. I posted a question to this thread about getting the AR488 code to support development of a GPIB device using GPIB secondary addressing in July 2020. The author of AR488 immediately replied and said he would be interested in working with me on my project! He created a new thread on eevblog to work on 'our' new project:
https://www.eevblog.com/forum/projects/tektronix-4924-tape-drive-emulator/msg3168024/#msg3168024

We started out with my logic analyzer trace of the “FIND” command to the 4924 and the results matched the 4924 service manual description of the sequence of GPIB transactions including use of the secondary address to encode the “FIND” command! We discussed my concept of using the FAT file system and encoding the 4051 tape header into the filename of each file – with the file number as the first part of the filename, like the tape header.

I began working through connecting a microSD card to the Arduino in addition to the pins needed for GPIB. I found the Arduino Nano board did not have enough IO pins and found the Pololu Microstar 328PB Micro had two more pins and used a ProMicro AR488 PCB to connect the Pololu 328PB Micro to GPIB with and Adafruit MicroSD adapter. I was able to load the AR488 program into this prototype and get EZGPIB to work.

I next added the SDFat v1.0 library to the AR488 program and added a prototype 4050 “TLIST” and “FIND” commands to the AR488 program! Both worked, but the Pololu 328PB Micro was running out of flash and memory space quickly.

I searched for a small Arduino board with 5V IO for direct compatibility with GPIB and more RAM and ROM for the Flash Drive program. The Arduino Mega might work but would be too large to mount directly to the 4050 GPIB connector. I found and ordered a Panduino 644 Narrow board which seemed to fit the requirements. There was also a version with the Atmega 1284 – with double the RAM and ROM of the 644 if we needed even more than double the RAM/ROM of the Arduino Nano or ProMicro.

My prototype with the 644-Narrow board connected to the AR488 ProMicro GPIB board and Adafruit MicroSD board worked with my TLIST and FIND commands. When I added OLD which ‘loaded’ an ASCII program file into the Realterm serial console – it was apparent we needed to switch to the 644-Narrow as the program no long fit in the Arduino Nano.

I quickly designed and ordered my first custom PCB – the Tektronix 4924 Tape Emulator which connects a 644-Narrow board to GPIB and includes the Pololu MicroSD adapter with 5V to 3.3V level shifter. I got this PCB assembled and was able to get it working first time with EZGPIB! However, my collaborator had already begun work to start fresh with a program designed for the Flash Drive device and eliminate all the AR488 controller code.

My collaborator purchased a 1284-Narrow board and his new “AR488-Store” program incorporated my prototype flash drive commands, so we abandoned the AR488 code and began working with his new program. I also assembled a second Flash Drive with 644-Narrow, my PCB with GPIB connector and Pololu MicroSD to my collaborator.

In July 2021 I was able to get the Flash Drive to successfully send ASCII strings to my 4052 using the INPUT @5: command! We were now on the way to having a working Flash Drive!. In addition – my collaborator purchased a 4051 in July 2021 and repaired it in August! Now we could both be able to run and debug the Flash Drive – and in addition be able to test all the Tektronix 4050 models: 4051, 4052 and 4054A!

We continued to use the eevblog forum to exchange progress reports until October 2021 when we moved to vcfed.org and used private messages.

Being able to test on all three different 4050 accelerated the development of the Flash Drive and by December 2021 both the 4051 and my 4052 were working – but my 4054A – with its different GPIB implementation using a TI 9914 GPIB IC was not working, particularly not with the READ and WRITE commands we added for R12 Graphics support.

I was contacted on vcfed.org by someone with access to another 4054A who asked if he could help debug the Flash Drive. This added another pair of eyes to the issue and we quickly made more progress and found the remaining bugs in the Flash Drive program.

2. 4050 GPIB Flash Drive Features

The GPIB Flash Drive has the following features:

1. Completely replaces 4050 internal tape drive for ALL program and data storage
2. Ready to run with 35 games and 33 R12/Fast Graphics pictures pre-installed on MicroSD card
3. MicroSD card provides Gigabytes of program data and storage
    a. Plug MicroSD into USB-MicroSD adapter to transfer program & data files to/from your PC
4. Faster access and loading of all files than internal tape
5. Stores each ‘tape’ in separate folder – 100’s of tapes can be stored on same Flash Drive
6. Compatible with all Tektronix 4051, 4052, 4052A, 4054 and 4054A computers
7. Supports all 4050 BASIC tape commands:
    a. FIND, MARK, KILL, OLD, BOLD, APPEND, PRINT, INPUT, READ, WRITE
8. Plugs into 4050 GPIB connector – no ROM backpack slots required
9. Flash Drive Micro-USB power cord included

3. Flash Drive Operation

Plug the Flash Drive into a USB 5V @ 1A power supply (not included) and then plug the Flash Drive into the GPIB connector on the back panel of your 4051, 4052 or 4054 computer.

Turn on the 4050 computer. The Flash Drive starts in the ROOT folder of the MicroSD card.

Flash Drive Command	Description
FIND@5: X	Opens file number X in the current folder
OLD@5:	Loads an opened flash drive file containing an ASCII program into 4050 memory
BOLD@5:	Loads an opened flash drive file containing a BINARY program into 4050 memory
SAVE@5:	Saves the current 4050 program to the opened flash drive as ASCII Program
BSAVE@5:	Saves the current 4050 program to the opened flash drive as BINARY Program
APPEND@5:	Appends opened flash drive file containing an ASCII program on current program
INPUT@5: X$, Y	Inputs ASCII data from an opened flash drive DATA file into the 4050
PRINT@5: X$, Y	Prints ASCII data into an opened flash drive DATA or NEW file from the 4050
READ@5: X, Y	Reads BINARY data from an opened flash drive DATA file into the 4050
WRITE@5: X, Y	Writes BINARY data into an opened flash drive DATA or NEW file
TYPE@5: X	Returns the type of the next BINARY data item in the current DATA file
MARK@5: 1, 1000	Creates a NEW file at the opened LAST file and then marks a new LAST file
KILL@5: X	Finds and marks file X as NEW
PRINT@5,9: "Directory"	Change to "Directory" folder
INPUT@5,19: A$	After a FIND@5: this will return that file header string into A$
PRINT@5,19: A$	After a FIND@5: this will replace the file header string with A$

Figure 2 - Flash Drive Commands

The Flash Drive uses the file name as the file header – like the Tape Drive. Easiest way to create a properly formatted filename is to copy an existing filename of the same type (ASCII PROGRAM for example) – changing the file number to the desired file number and editing the comment field. Be careful to not change the location of the ASCII/BINARY or PROGRAM/DATA field. You can test whether the created file name is correct by using the Main Menu "TLIST" command on that folder.

Here is the format of every Flash Drive filename:

```
    1      ASCII   PROG Root Menu        2826
    |      |       |    |              | |_____File-size starts at      38 and is automatic
    |      |       |    |______________|_______Comment/filename starts @ 21 and ends at 36
    |      |       |___________________________PROG or DATA starts at    16
    |      |___________________________________ASCII or BINARY starts at 8
    |__________________________________________File number starts at     1
```

Figure 3 - Flash Drive Filename Format

This format is used by the Flash Drive to properly access the data requests from 4050 BASIC based on the file type. You may notice these character start positions are 1 less than on the tape header locations published in the 4050 programming reference. In that document the first character is always a space which is illegal in FAT and other filesystems. Our observation is that 4050 BASIC never requests the file header from a 4924 GPIB tape drive. The 4924 Tape Drive service manual indicates the 4924 uses the file header to determine how to operate on 4050 reads and writes, so that is how we designed the Flash Drive operation.

There is no flash drive command to create or delete folders on the MicroSD. That is easily done by unplugging the flash drive MicroSD card (push gently and remove card from adapter) and plugging it into a USB to MicroSD card reader (not supplied). Plug the MicroSD card reader into your PC and use the PC to create or delete a folder from the Flash Drive MicroSD card. Folder names should be limited to 12 characters.

SECRET files are NOT supported in the Flash Drive system.

4. Flash Drive hardware

The Flash Drive is comprised of three circuit boards:

1. Controller: Pandauino 644-Narrow or 1284-Narrow running the Flash Drive Arduino code
2. MicroSD Adapter: Pololu #2587 MicroSD adapter with Level Shifter. Uses any MicroSD card
3. Flash Drive GPIB Interface: connects Controller, MicroSD adapter and GPIB connector

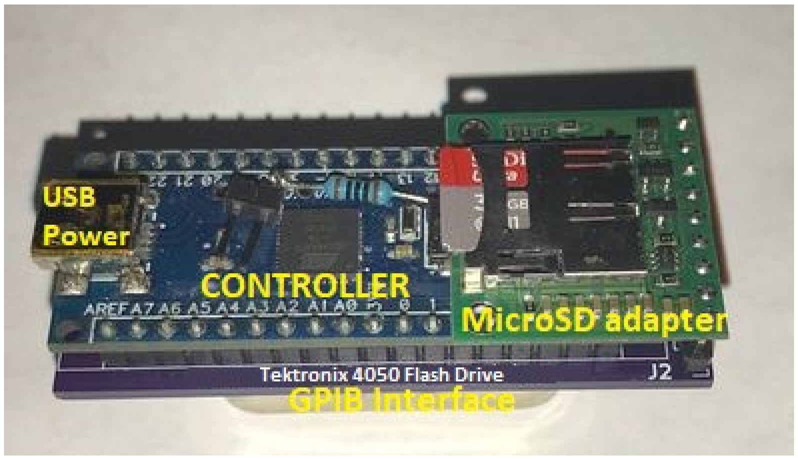

Figure 4 - Flash Drive hardware

5. Micro SD card files

The default folders and files on the Flash Drive MicroSD card are preinstalled.

Any updates to these files or folders will be uploaded to

https://github.com/mmcgraw74/Tektronix-4051-4052-4054-Program-Files/tree/master/Flash_Drive

as a zip file. Easiest way to freshen or restore the Flash Drive folders and files is to delete all the folders and files from the MicroSD card using your PC and then unzip the new folders and files to the MicroSD card.

6. Flash Drive software

The Flash Drive software is preloaded into the Flash Drive controller. Updates to the Flash Drive software will be posted in this folder: https://github.com/mmcgraw74/Tektronix-4051-4052-4054-Program-Files/tree/master/Flash_Drive/AR488_Store

The Flash Drive software can be updated from a PC with Arduino IDE installed with the MightyCore link added to Board Manager and selecting your controller board - ATmega644 or ATmega1284 - using the included Micro-USB cable.

https://mcudude.github.io/MightyCore/package_MCUdude_MightyCore_index.json