Pimp my Gotek

Contents

Summary

I started with a Gotek I purchased from Amigakit back in 2017. I'm not sure what version of software was included, but it did not accept updating firmware automatically from the flash drive as newer firmware versions support. This model also has the old display which only showed 3 numbers and has the two buttons for navigation.

Given that so many of us re-house the drive in a 3d printed bracket and screens and rotary dials are so cheap, knowing how to modify your own Gotek could be beneficial for those wanting to save a few dollars.

I found great information and tutorials at the Flash Floppy Wiki site https://github.com/keirf/flashfloppy/wiki/Firmware-Programming.

Please note, this information is targeted solely at the STM32 model of Gotek that I modified and documented here. If you are not familiar with the models, the Flash Floppy wiki can help you out.

Programming a "STM32" model Gotek using the new STM32CubeProgrammer

Getting started

I understand that this method of flashing firmware is similar to what you would do with un unmodified Gotek drive. Most of what I did reflects the information found on the Flash Floppy wiki with one notable exception. The tools referenced on the site are deprecated now. While some tools are still available. I decided to take the route of downloading and using the new tool with the new UI. This section, while overlapping with information from the Wiki, is intended to be a quick start to using the new flash tool, STM32CubeProgrammer. This focuses on flashing the firmware and is not a complete tutorial on that tool!

Download and install the software

The software is available at https://www.st.com/en/development-tools/stm32cubeprog.html.

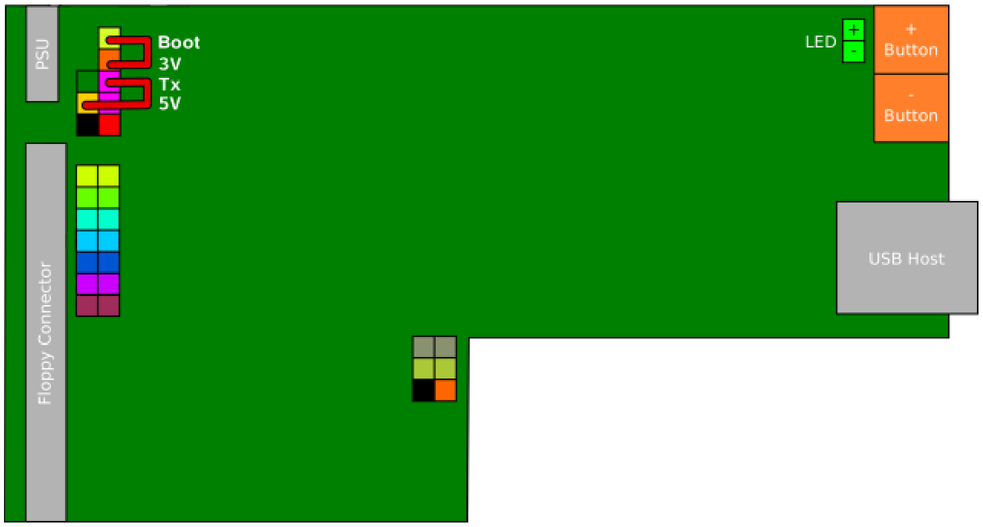
This will install both the flash programmer, the USB driver, and the STM32TrustedPackageCreator (not used here)

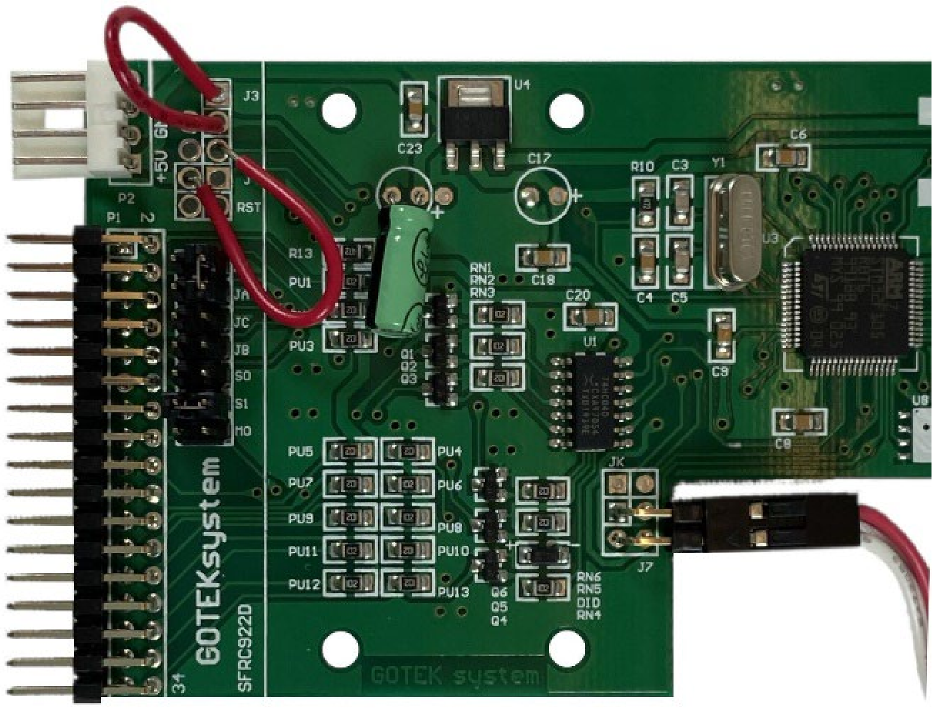
Unzip the download and run the setup, accepting defaults.

Set the jumpers on your Gotek

For a complete guide, see https://github.com/keirf/flashfloppy/wiki/Firmware-Programming.

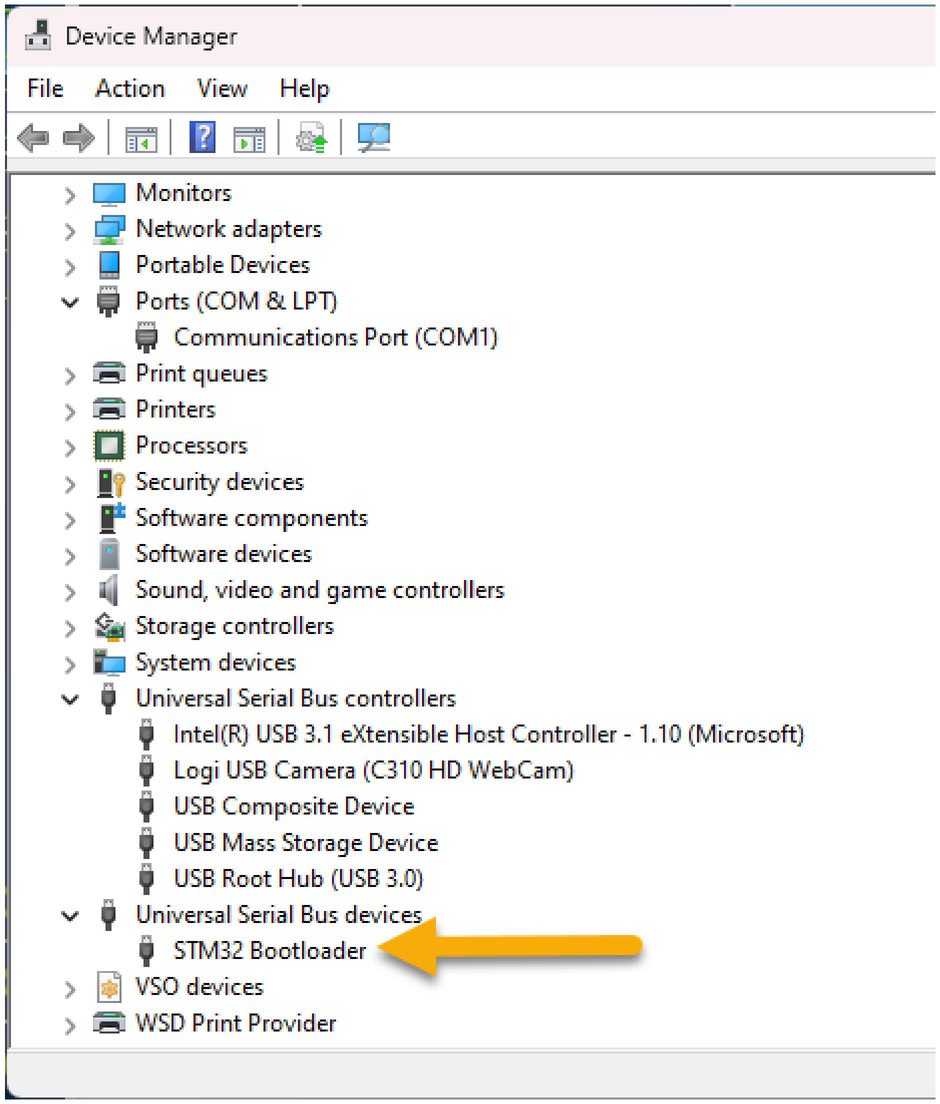
For reference, I have the jumper settings for my purpose copied here. The jumper area is typically not populated with pin headers. You can choose you own way of jumping these connections. I decided to add pin headers since I have plenty available with jumper wires to easily connect and disconnect multiple times as I learn this new software.





Connect the Gotek to Windows

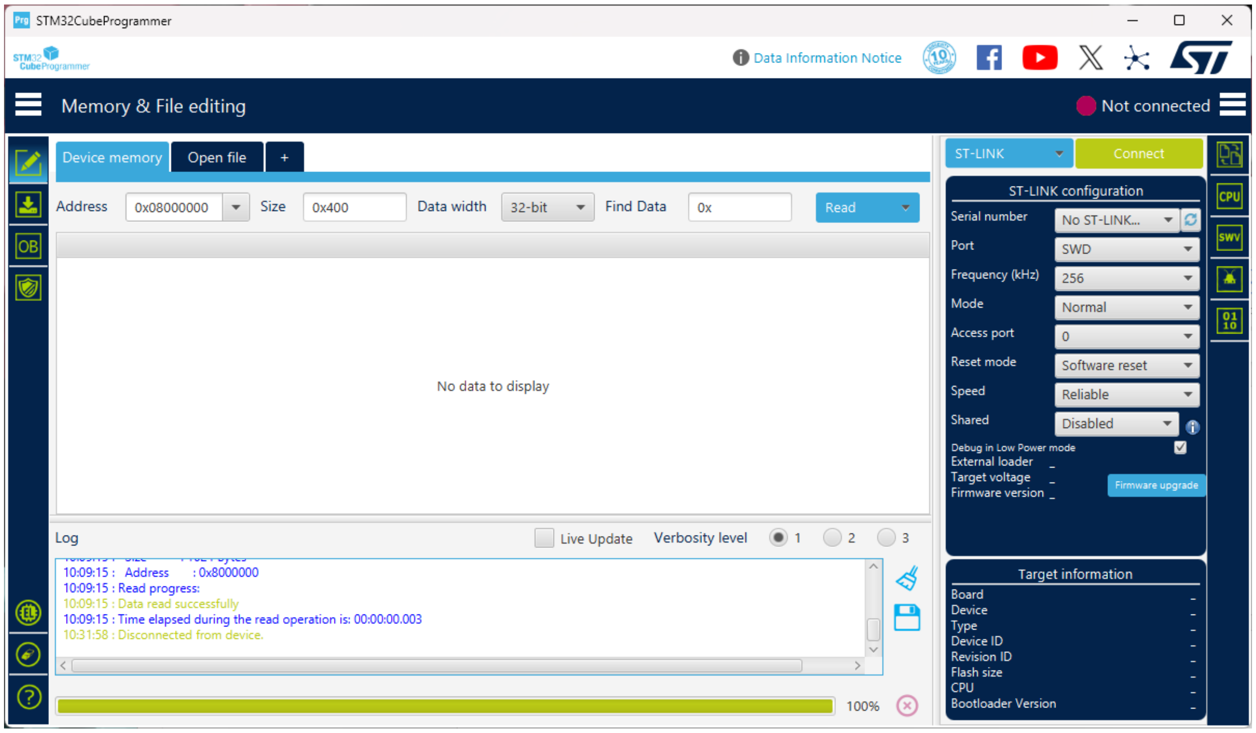
On my computer, connecting the properly configured/Jumped Gotek to Windows resulted in the USB bus sending a couple of unknown device notifications (each time I plug in) before settling down with the correct drive applied. You know it's completed connection properly when you see the STM Bootloader in the Device Manger as pictured below.



Monitors
Network adapters
Portable Devices
Ports (COM & LPT)
Communications Port (COM1)
Print queues
Printers
Processors
Security devices
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
Intel(R) USB 3.1 eXtensible Host Controller - 1.10 (Microsoft)
Logi USB Camera (C310 HD WebCam)
USB Composite Device
USB Mass Storage Device
USB Root Hub (USB 3.0)
Universal Serial Bus devices
STM32 Bootloader
VSO devices
WSD Print Provider

Launch the STM32CubeProgrammer

When you click on the STM32CubeProgrammer icon on your Windows Desktop, you will be greeted with the screed below. You will be on the Memory & File editing tab, which is where you want to be. You will notice near the top right that we are “Not Connected”.



Address 0x08000000 Size 0x400 Data width 32-bit Find Data 0x

Verbosity level [✓ 1] 2 3

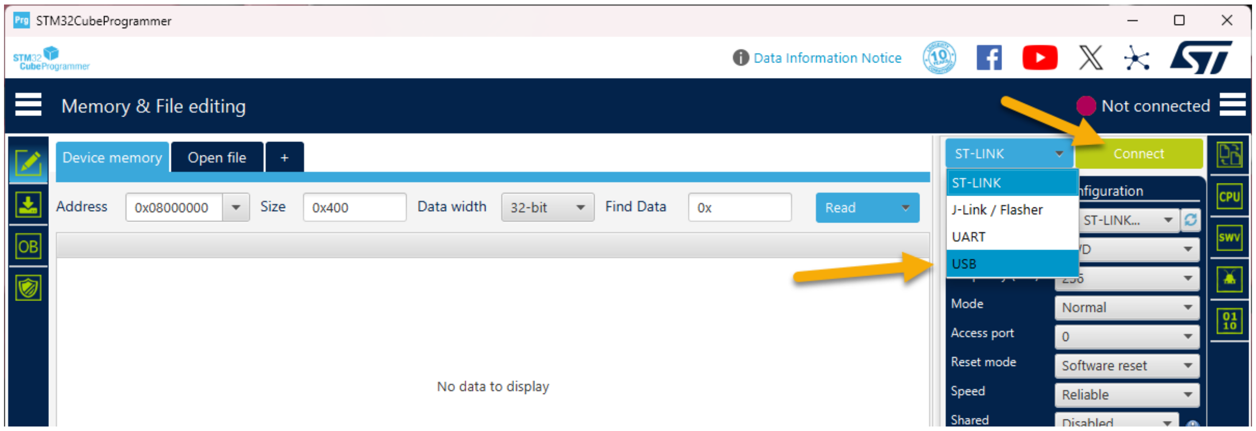
```
10:09:15 :   Address      : 0x8000000
10:09:15 : Read progress:
10:09:15 : Data read successfully
10:09:15 : Time elapsed during the read operation is: 00:00:00.003
10:31:58 : Disconnected from device.
```

Serial number	No ST-LINK...
Port	SWD
Frequency (kHz)	256
Mode	Normal
Access port	0
Reset mode	Software reset
Speed	Reliable
Shared	Disabled
Debug in Low Power mode	✓
External loader	_
Target voltage	_
Firmware version	_

Board	-
Device	-
Type	-
Device ID	-
Revision ID	-
Flash size	-
CPU	-
Bootloader Version	-

Connect to the Gotek

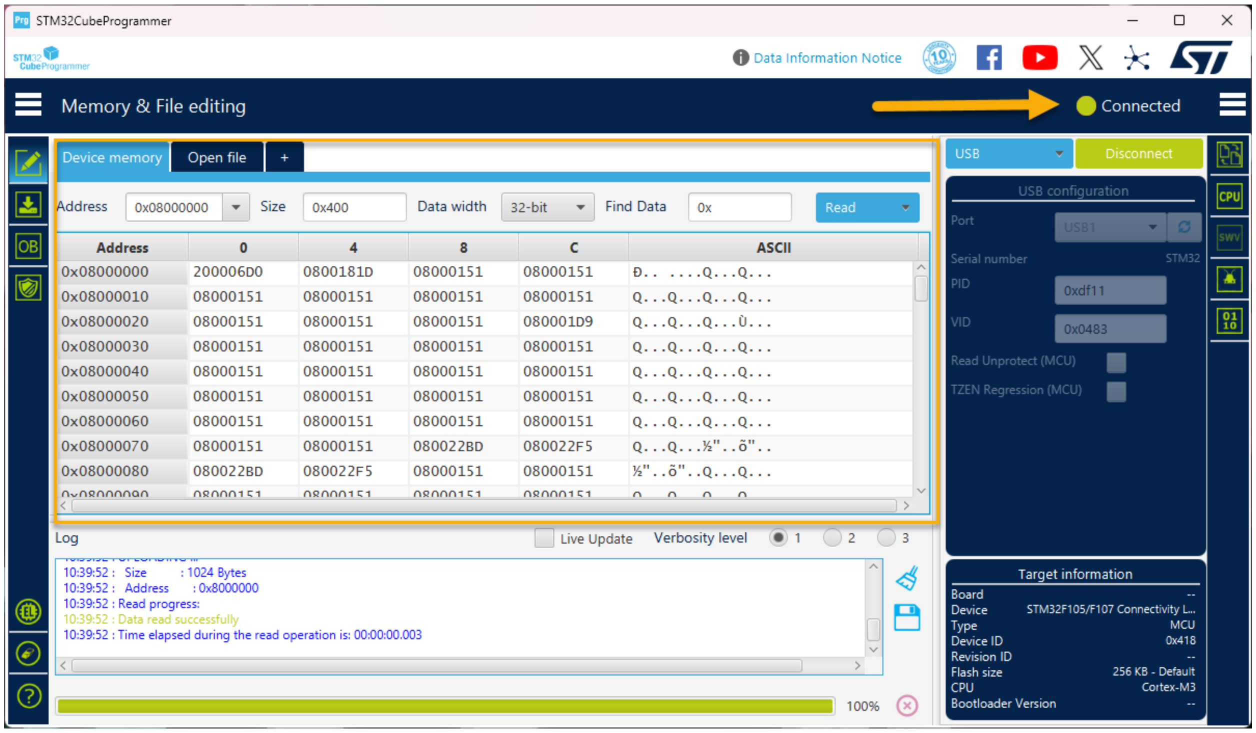
We need to connect to the Gotek now. Before we do that, we must first switch from ST-Link to USB in the dropdown to the left of the Connect button.



Address: 0x08000000 Size: 0x400 Data width: 32-bit Find Data: 0x

Mode: Normal
Access port: 0
Reset mode: Software reset
Speed: Reliable
Shared: Disabled

After selecting USB, click connect and the "Not connected" message will change to "Connected". The program will also display a dump of the device memory.



Address: 0x08000000 Size: 0x400 Data width: 32-bit Find Data: 0x

```
Address       0          4          8          C          ASCII
0x08000000    200006D0   0800181D   08000151   08000151   Đ..  ....Q...Q...
0x08000010    08000151   08000151   08000151   08000151   Q...Q...Q...Q...
0x08000020    08000151   08000151   08000151   080001D9   Q...Q...Q...Ù...
0x08000030    08000151   08000151   08000151   08000151   Q...Q...Q...Q...
0x08000040    08000151   08000151   08000151   08000151   Q...Q...Q...Q...
0x08000050    08000151   08000151   08000151   08000151   Q...Q...Q...Q...
0x08000060    08000151   08000151   08000151   08000151   Q...Q...Q...Q...
0x08000070    08000151   08000151   080022BD   080022F5   Q...Q...½"..õ"..
0x08000080    080022BD   080022F5   08000151   08000151   ½"..õ"..Q...Q...
```

Port: USB1
Serial number: STM32
PID: 0xdf11
VID: 0x0483
Read Unprotect (MCU)
TZEN Regression (MCU)

Log — Verbosity level: [✓ 1] 2 3

```
10:39:52 :  Size        : 1024 Bytes
10:39:52 :  Address     : 0x8000000
10:39:52 : Read progress:
10:39:52 : Data read successfully
10:39:52 : Time elapsed during the read operation is: 00:00:00.003
```

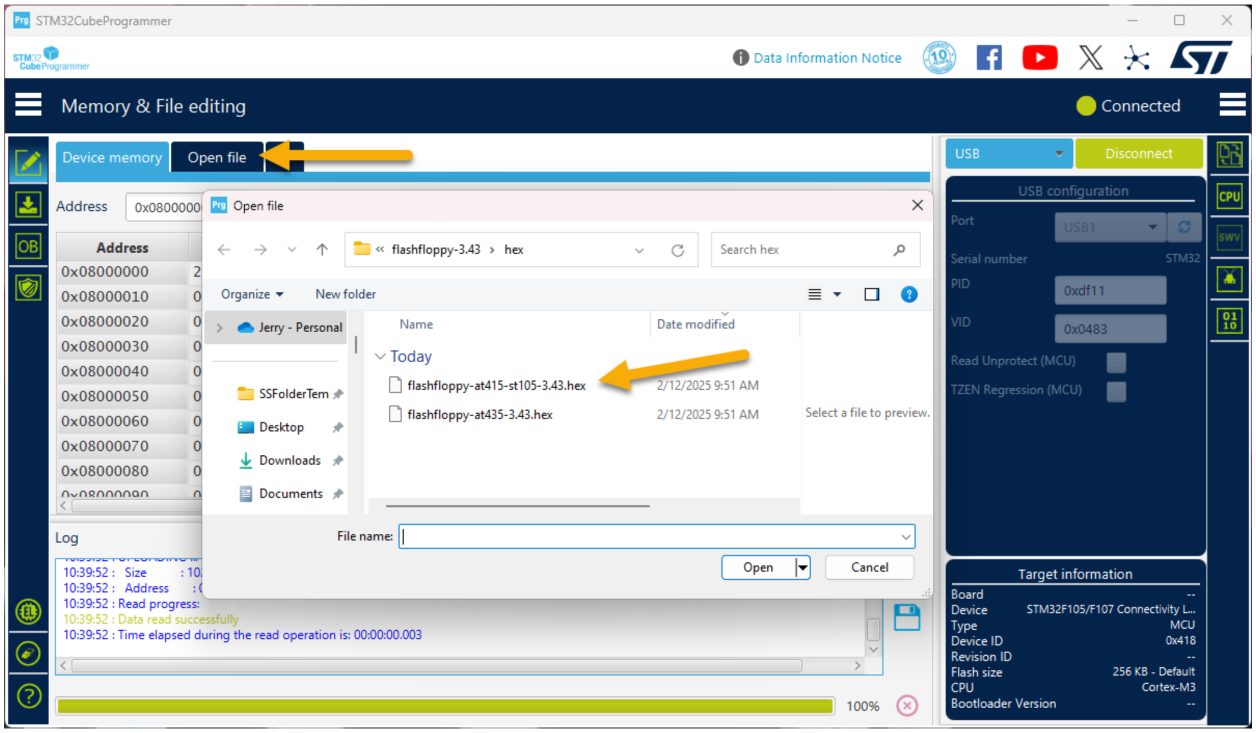
100%

Target information	
Board	--
Device	STM32F105/F107 Connectivity L...
Type	MCU
Device ID	0x418
Revision ID	--
Flash size	256 KB - Default
CPU	Cortex-M3
Bootloader Version	--

Load the firmware .hex file

The firmware is available on GitHub site https://github.com/keirf/flashfloppy under releases. You will need to go there, download and unzip the file prior to this step.

Now you can click the Open file tab which launches a dialog to select a file. Navigate to the location of the firmware. Select the .hex file for the firmware you want to upload.



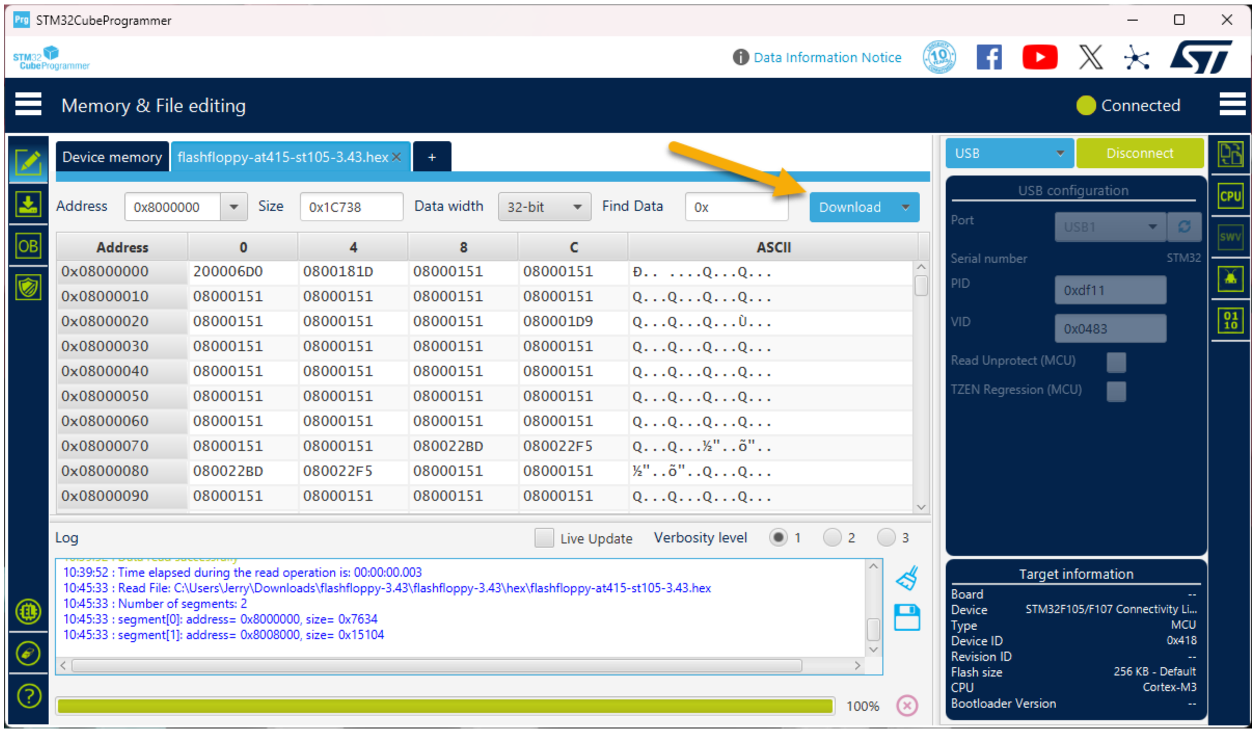
Address
0x08000000
0x08000010
0x08000020
0x08000030
0x08000040
0x08000050
0x08000060
0x08000070
0x08000080

```
10:39:52 :  Size       : 10
10:39:52 :  Address    : (
10:39:52 : Read progress:
10:39:52 : Data read successfully
10:39:52 : Time elapsed during the read operation is: 00:00:00.003
```

Target information	
Board	--
Device	STM32F105/F107 Connectivity L...
Type	MCU
Device ID	0x418
Revision ID	--
Flash size	256 KB - Default
CPU	Cortex-M3
Bootloader Version	--

Click Download

You will see your .hex file is not in the highlighted tab. The default values are good. Just click download to flash the device. The program will show it's progress.



Address 0x8000000 Size 0x1C738 Data width 32-bit Find Data 0x

Address	0	4	8	C	ASCII
0x08000000	200006D0	0800181D	08000151	08000151	Ð.. ....Q...Q...
0x08000010	08000151	08000151	08000151	08000151	Q...Q...Q...Q...
0x08000020	08000151	08000151	08000151	080001D9	Q...Q...Q...Ù...
0x08000030	08000151	08000151	08000151	08000151	Q...Q...Q...Q...
0x08000040	08000151	08000151	08000151	08000151	Q...Q...Q...Q...
0x08000050	08000151	08000151	08000151	08000151	Q...Q...Q...Q...
0x08000060	08000151	08000151	08000151	08000151	Q...Q...Q...Q...
0x08000070	08000151	08000151	080022BD	080022F5	Q...Q...½"..õ"..
0x08000080	080022BD	080022F5	08000151	08000151	½"..õ"..Q...Q...
0x08000090	08000151	08000151	08000151	08000151	Q...Q...Q...Q...

Live Update   Verbosity level [✓ 1] 2 3

```
10:39:52 : Time elapsed during the read operation is: 00:00:00.003
10:45:33 : Read File: C:\Users\Jerry\Downloads\flashfloppy-3.43\flashfloppy-3.43\hex\flashfloppy-at415-st105-3.43.hex
10:45:33 : Number of segments: 2
10:45:33 : segment[0]: address= 0x8000000, size= 0x7634
10:45:33 : segment[1]: address= 0x8008000, size= 0x15104
```

100%

Port	USB1
Serial number	STM32
PID	0xdf11
VID	0x0483
Read Unprotect (MCU)	
TZEN Regression (MCU)	

Target information

Board	--
Device	STM32F105/F107 Connectivity Li...
Type	MCU
Device ID	0x418
Revision ID	--
Flash size	256 KB - Default
CPU	Cortex-M3
Bootloader Version	--

Verify the result

Disconnect the device from Windows and remove the jumpers. Install it in your machine and test it out.

Adding a rotary encoder

Getting started

Once again, the information on the Flash Floppy wiki gives you all the information to get going on this project.

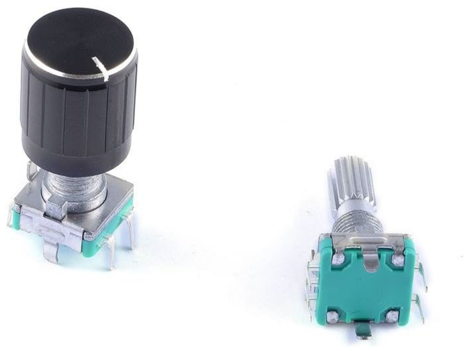
https://github.com/keirf/flashfloppy/wiki/Hardware-Mods#rotary-encoder

Purchase the Encoder

Finding the proper part is sometimes the most challenging aspect of the modification. There are often many choices and us novice modders have to take our best educated guess to get the right parts! I found the following at Amazon which worked well. The 5 pack was $9.

5Pcs 360 Degree Rotary Encoder Code Switch Digital Potentiometer with Push Button 5 Pins and Knob Cap for Arduino (Pack of 5) CYT1100

https://www.amazon.com/dp/B07DM2YMT4?psc=1&ref=ppx_pop_dt_b_product_details



Wiring the Encoder

Finding the part was only the first challenge. The part did not come with any pinout information so I had to hunt that down as well and hope for the best. I was able to use my meter to verify the switch pins, so I felt confident the remaining pinout was correct. I found that the connection information at the Flash Floppy site matched what I found independently, so I believe this pinout is probably common, or even standard.

I used the diagram from the flash floppy site to wire my rotary encoder to the Gotek. Note, there are newer models that have a dedicated connection point for encoders that is much more convenient that this. It's all covered on the Flash Floppy site.

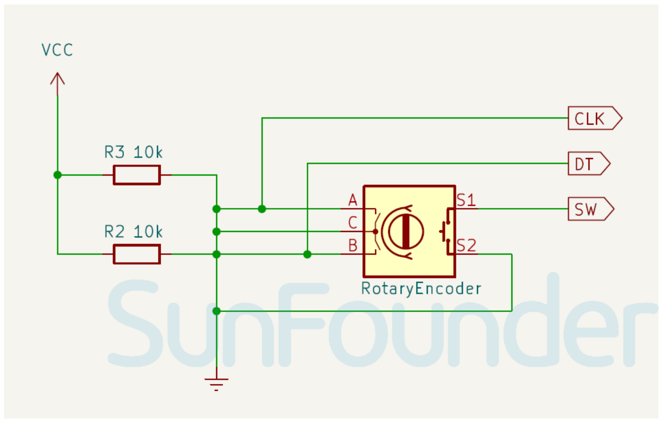


Figure 1- Pinout found online

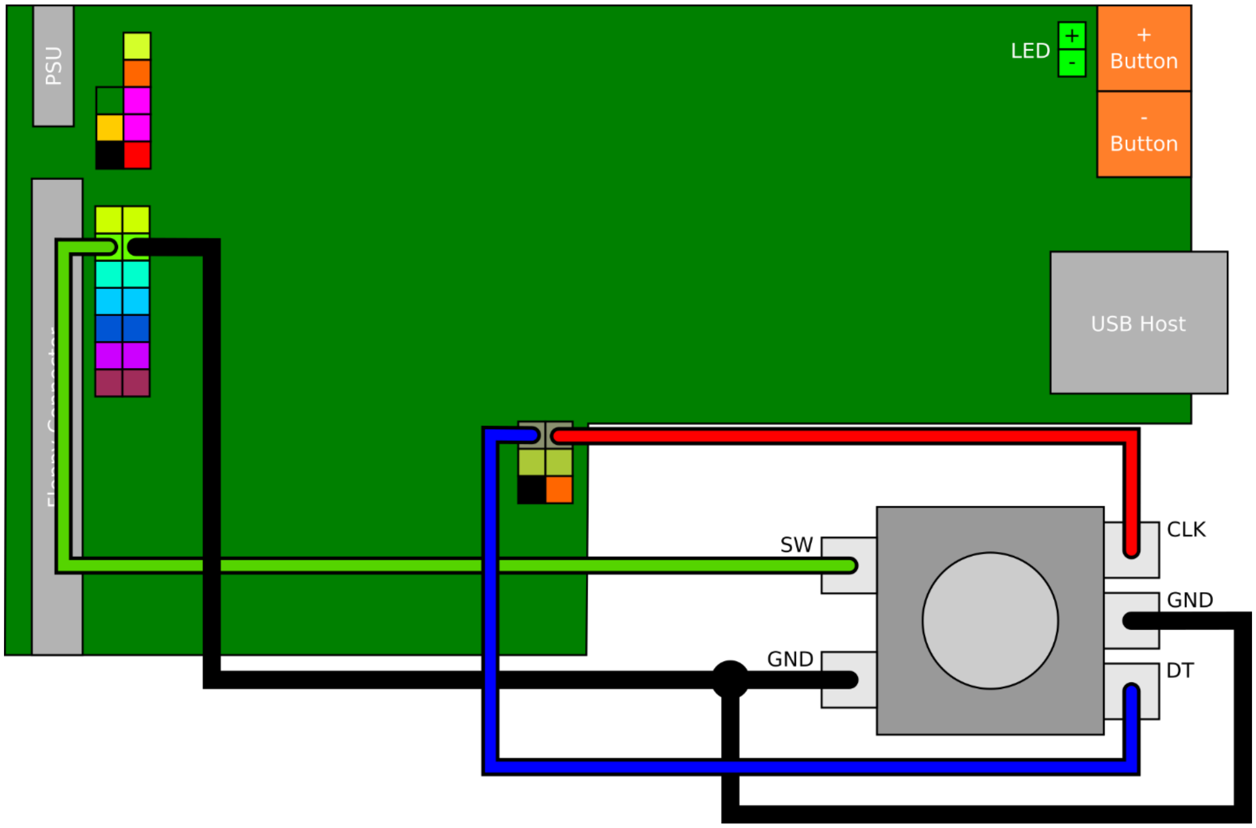


Figure 2- Pinout/Connection diagram at the Flash Floppy site

I soldered some test leads to the rotary encoder which had the pin-header connections attached on one end to facilitate easily connecting the encoder to the Gotek. It worked perfect first time!

Adding a new OLED display

Getting started

If you have a newer firmware that supports the new screen, this is as easy as connecting the new screen to the connections already present for the "7-Segment" display and updating your Flash Floppy configuration file.

https://github.com/keirf/flashfloppy/wiki/Hardware-Mods#oled-display.

Selecting the display

These displays are fairly common for people creating Arduino, Pi, Pi Pico projects. You will want and "I2C" style display with VCC, GND, SCL, and SDA connections.

The .91 inch OLEDs are popular for those who want to keep the original Gotek case. There is a 3d printed bracket that incorporates in the stock case to hole this display nicely in the original display hole. The exact model I purchased was:

MakerFocus 2pcs I2C OLED Display Module 0.91 Inch I2C SSD1306 OLED Display Module Blue I2C OLED Screen Driver DC 3.3V~5V for Arduino

https://www.amazon.com/dp/B0761LV1SD?ref_=ppx_hzsearch_conn_dt_b_fed_asin_title_3

This one is already unavailable!! But there are others. They appear to cost around $4 each now ($8 for a two pack).

I also purchase and opted to go with a slightly larger screen. I expect to re-case this eventually into another 3d printed bracket. For now I made a few modifications to the stock casing to test it out until I find or design the new bracket.

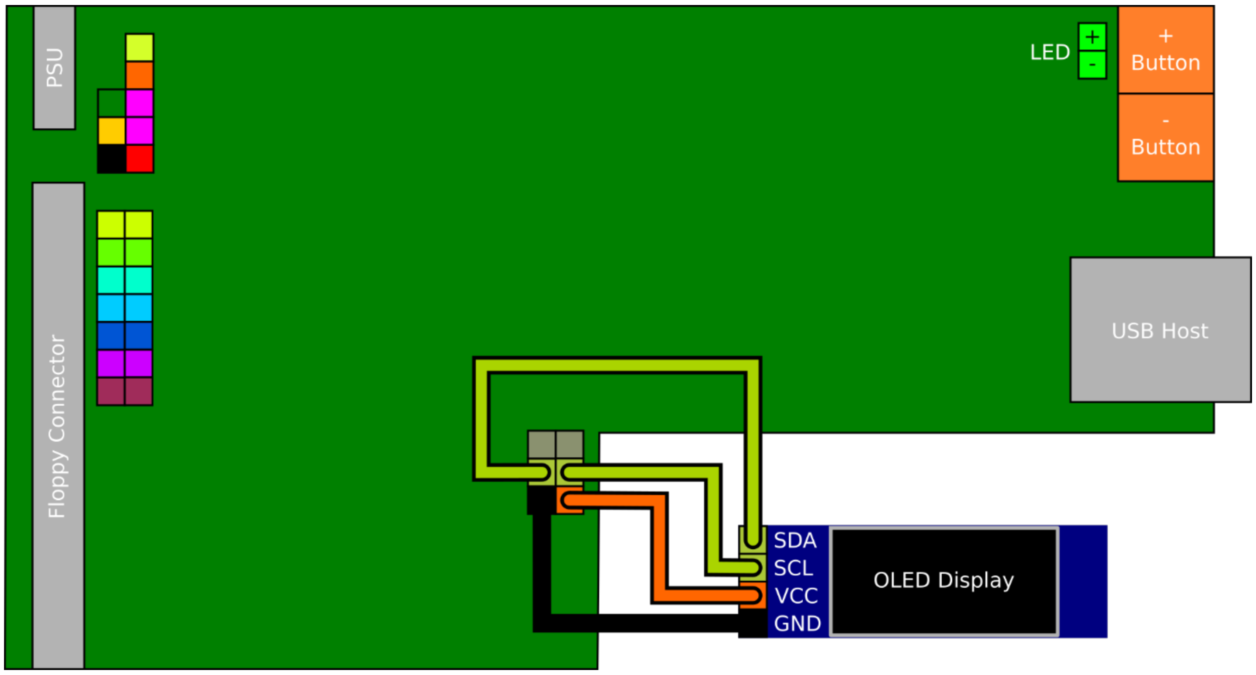
3Pcs 0.96 Inch OLED Display 3-Colors 128x64 SSD1306 SPI I2C IIC Serial LCD Screen Module Board with 4 Pins Soldered and 10 Dupont Cable Work Great with Arduino for Raspberry Pi

https://www.amazon.com/dp/B0BK4QVR1F?ref_=ppx_hzsearch_conn_dt_b_fed_asin_title_3

This sample pack of three screens was $10, still available, and each screen has a different color scheme to try out.

Connecting the display

The displays will have 4 connections: VCC, GND, SCL, SDA. Just follow the Flash Floppy diagram and you should be fine!



Updating the Flash Floppy Config (ff.cfg)

The FF.CFG is a very easy way to set/teak your Flash Floppy device behavior. This controls many different aspects of the device such as the floppy volume (if you have a speaker attached), scroll speed of disk images with long names, delay before mounting a disk, and more. We will look at the settings around the display.

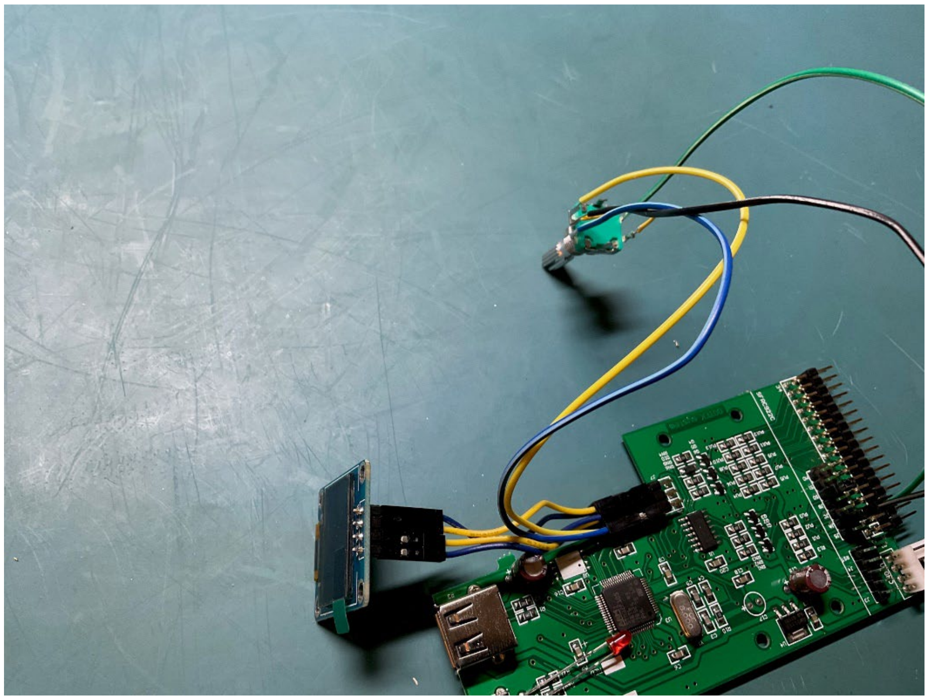
Here is a copy of the DISPLAY section of my FF.CFG with settings highlighted that I changed to best accommodate my display. Note, the # denotes a comment and is NOT a configuration setting.

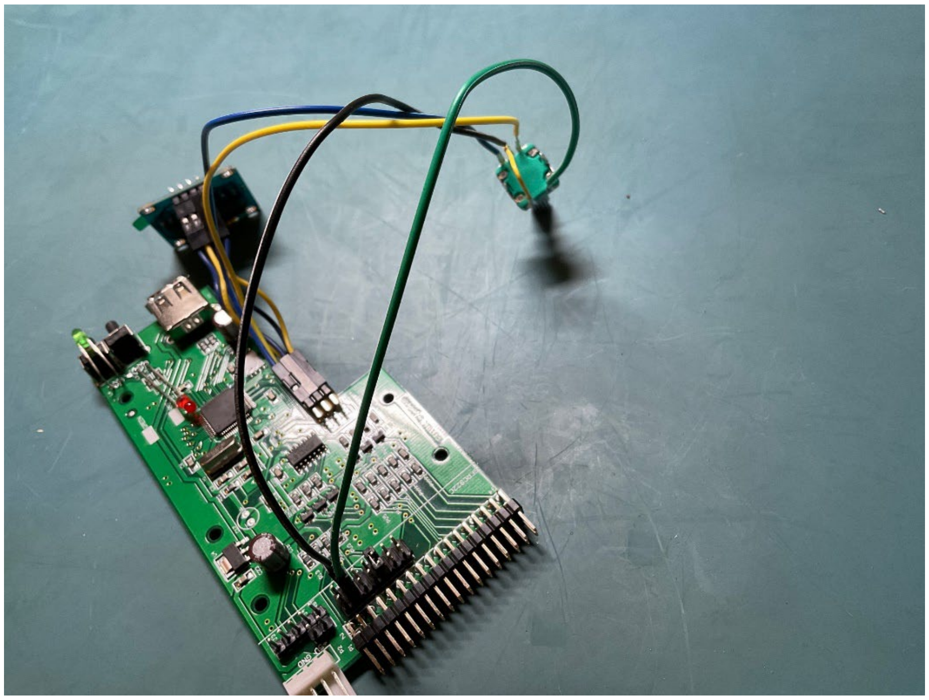
I only changed 2 values. The display type (display-type = oled-128x64), and what is displayed on each line (display-order = 1,0d,3). Really, based on the display type, I could have just accepted the default display-order for that type. I just wanted to mix it up a bit.

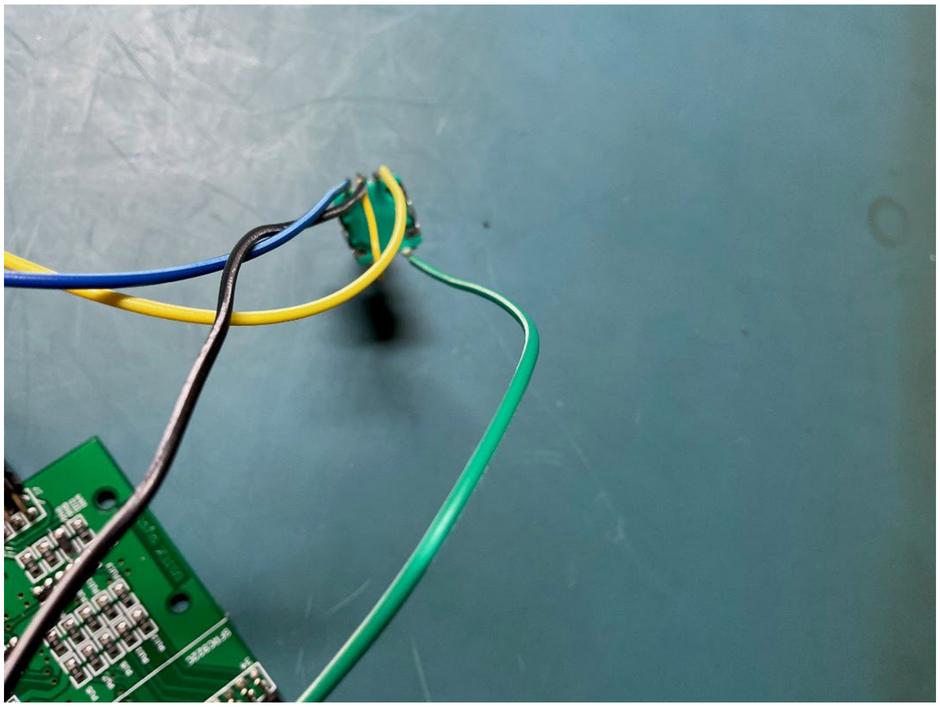
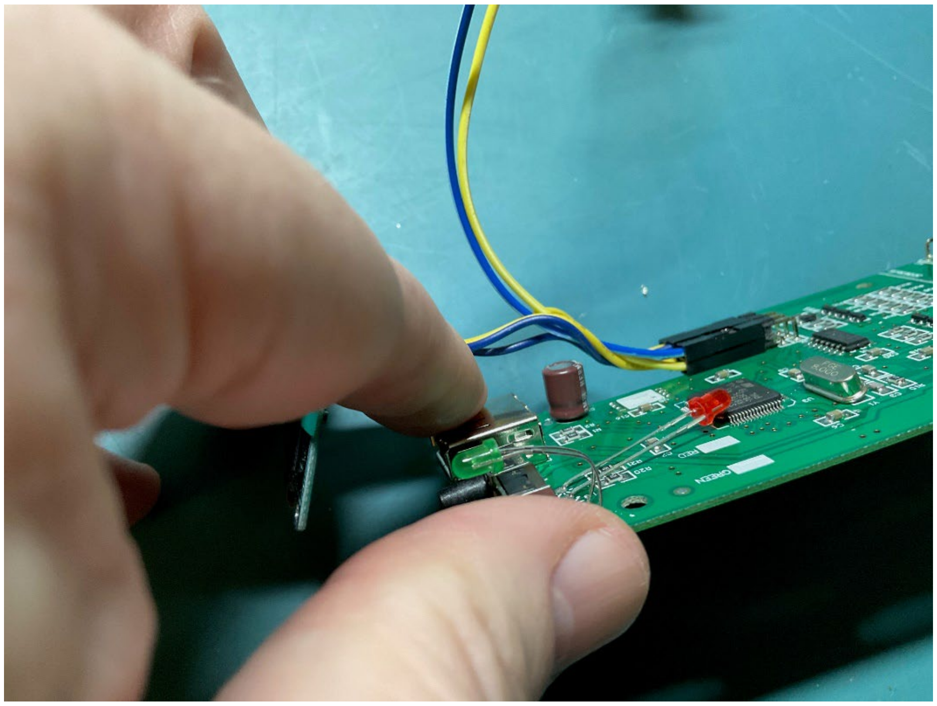
```
##

## 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
#  -hflip:     OLED view is flipped horizontally
#  -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
#  -slow:      Run I2C bus slower (use this if OLED regularly blanks/corrupts)
# Values: auto | lcd-CCxRR | oled-128xNN[-rotate][-narrow[er]]...
display-type = oled-128x64

# 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 = 144

# Text height and arrangement on LCD/OLED and on OSD, respectively.
# Comma-separated list, one entry per display row, top down.
# Each list item is a digit plus optional height specifier: [0-7][d]
#  content-row: 0-3 = specified content, 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 = 1,0d,3
osd-display-order = default

# OSD text columns. This is currently respected only when no LCD/OLED is found.
# Values: 16 <= N <= 40
osd-columns = 40

# 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 when there is 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 milliseconds per update
# Values: 100 <= N <= 65535
display-scroll-rate = 200
```

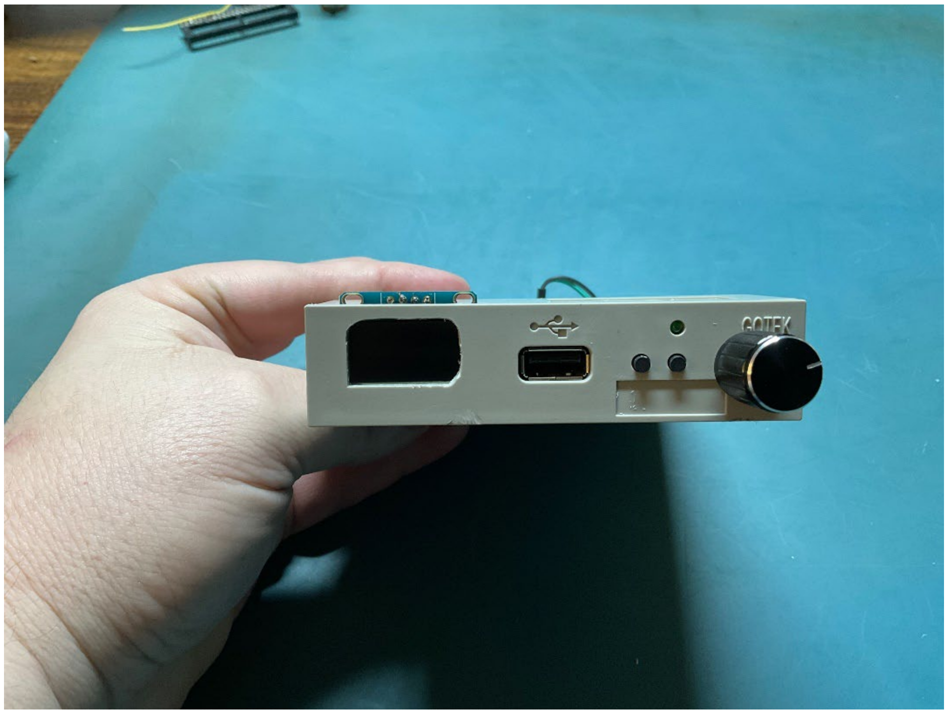
```
# LCD/OLED pause time at start/end of scroll, in milliseconds

# 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
```

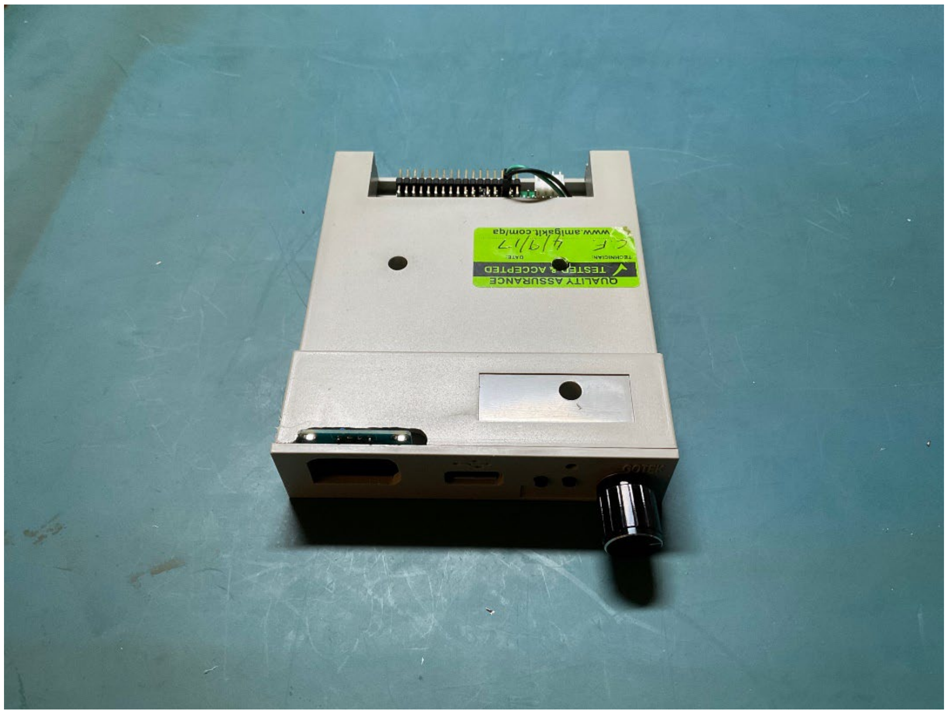
My Pics

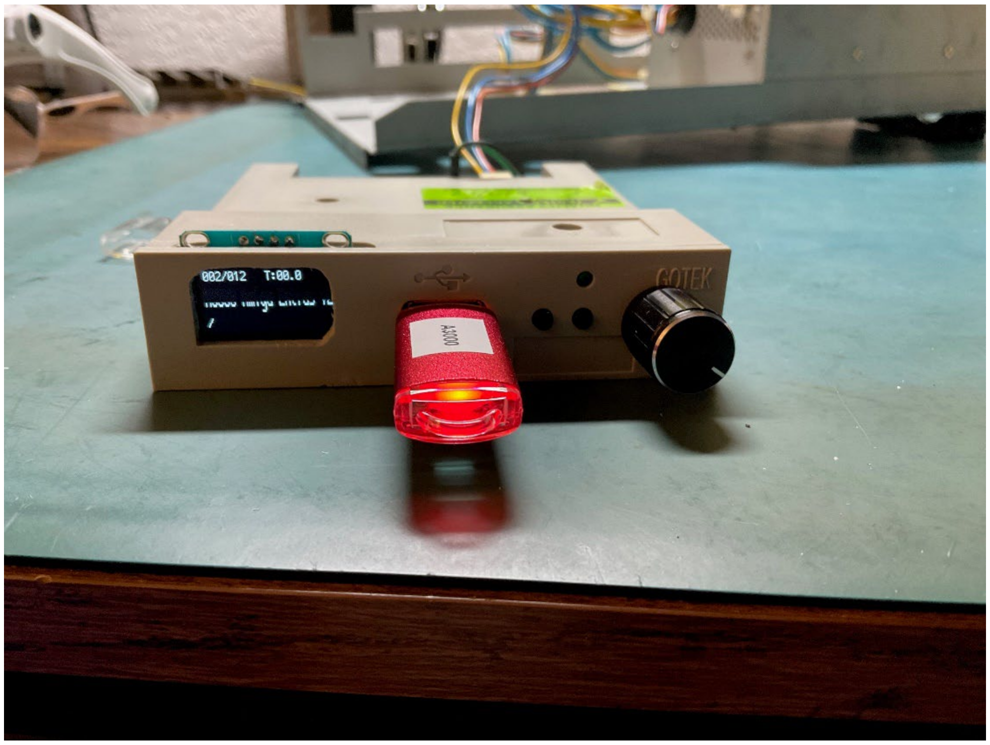






RED
GREEN


GOTEK





The screen works, the refresh was caught by the camera. The display has the status on top (file number x of y selected and disk Track information), disk image name in the middle in large text, and the subfolder (root - / in this case) on the bottom.