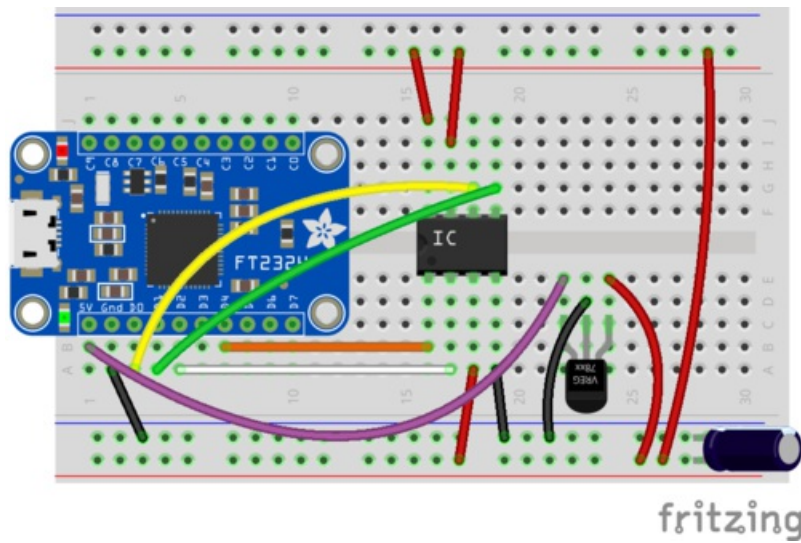


Programming SPI flash with an FT232H breakout

Created by Dean Miller



Last updated on 2018-08-22 04:01:16 PM UTC

Guide Contents

Guide Contents	2
Overview	3
SPI Flash Standardization	3
If you want to use an microcontroller...	3
USB to SPI gateway	3
Adafruit FT232H Breakout - General Purpose USB to GPIO+SPI+I2C	4
3.3V 250mA Linear Voltage Regulator - L4931-3.3 TO-92	4
10uF 50V Electrolytic Capacitors - Pack of 10	4
SMT Test Socket - SOIC-8 Narrow Breakout	4
SMT Breakout PCB for SOIC-8, MSOP-8 or TSSOP-8 - 6 Pack!	5
Download and compile	5
Wiring	5

Overview

In your journey hacking, modding and making electronics you will bump into many a FLASH chip. Often times these store program memory, settings, data files etc. Some microcontrollers have built-in flash, but an external flash chip allows for field-updating. Also, fitting a large Flash chip inside a micro or FPGA can increase cost a lot, where-as having it be external lets the customer pick exactly the right size they need. They can range from 256 bytes up to 16 MBytes or even more!

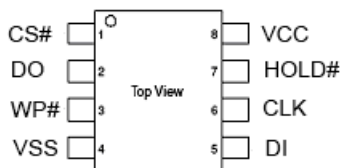
SPI Flash Standardization

The good news is that just about every 8-pin Flash chip has a standard pinout and SPI interface, these tend to have the number **25** somewhere in the beginning of the part number. There are also ones that are only I2C - these will have the number **24** somewhere in the part number. This page is just about SPI flash.

You can pick up any kind/size SPI flash you like over at Digkey (<https://adafru.it/xhd>)

As you can see, SPI flash part numbers tend to start with "MX25" or "W25" or "AT25" or "SST25" etc. The first two letters are the manufacturer name.

The rest of the part number will also contain a 3 digit number that indicates the size in 'kilobits' or 'megabits'. E.g. **010** is 1-Megabit (128 KByte), **080** is 8-Megabit (1 MByte), **016** is 16-Mbit/2MByte. For smaller sizes, all 3 digits are used. E.g. **256** is 256-Kbit (32KByte),



Not only do they have a standard pinout, but there is also a standard-ish set of commands you can use to read and write the data. So, if your flash memory chip has this pinout, chances are you can use a standard command set to program it.

If you want to use an microcontroller...

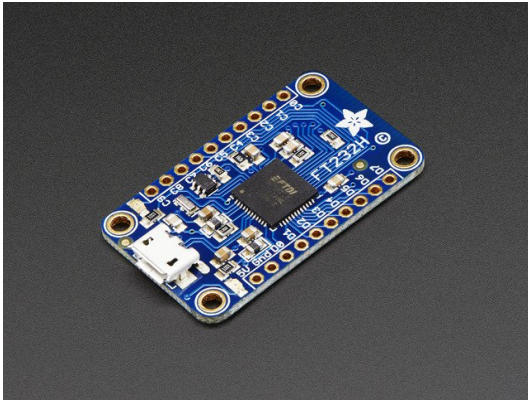
If you have an Arduino or compatible, check out our libraries (https://github.com/adafruit/Adafruit_SPIFlash (<https://adafru.it/wbt>) or https://github.com/adafruit/Adafruit_TinyFlash (<https://adafru.it/xhe>)) on how to read/write to/from these in a microcontroller

HOWEVER

you are here because you do not want to mess with an Arduino - your file that you want to read/write is on your computer, and its big so getting it into a microcontroller is a pain. Instead, in this guide we will use an FT232H as a 'gateway' so that you can use any computer and read/write the SPI flash through the command line.

USB to SPI gateway

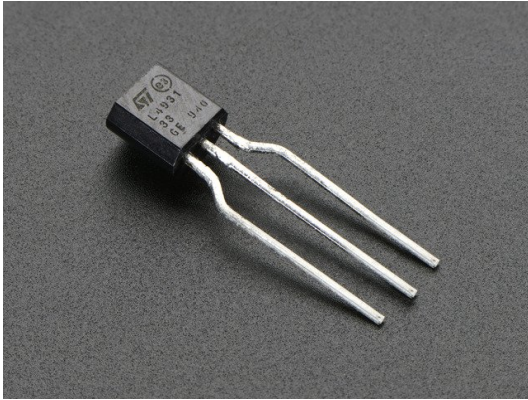
You will need the FT232H breakout, and if you have a chip in a package that is not breadboard-able right away you will need a socket or an adapter board.



Adafruit FT232H Breakout - General Purpose USB to GPIO+SPI+I2C

\$14.95
IN STOCK

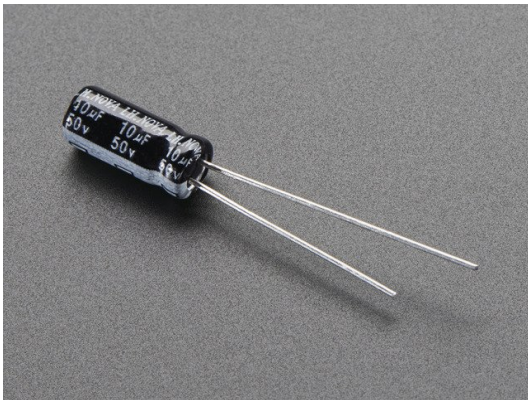
ADD TO CART



3.3V 250mA Linear Voltage Regulator - L4931-3.3 TO-92

\$0.95
IN STOCK

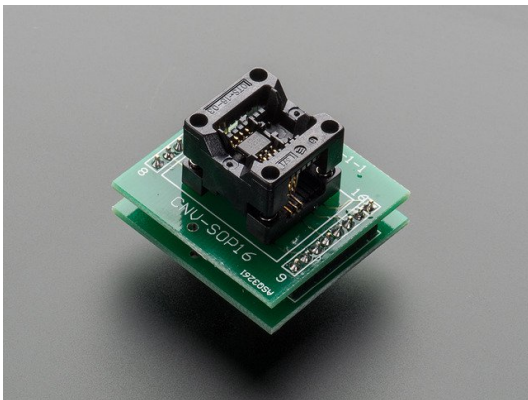
ADD TO CART



10uF 50V Electrolytic Capacitors - Pack of 10

\$1.95
IN STOCK

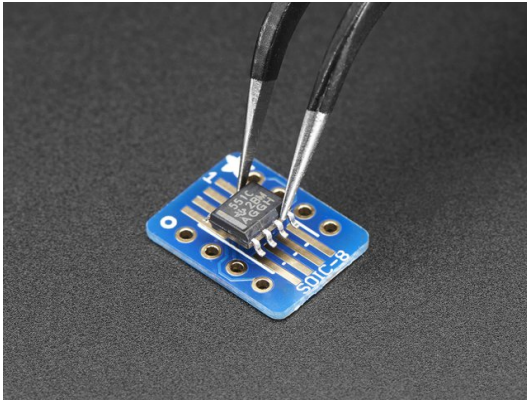
ADD TO CART



SMT Test Socket - SOIC-8 Narrow Breakout

\$14.95
IN STOCK

ADD TO CART

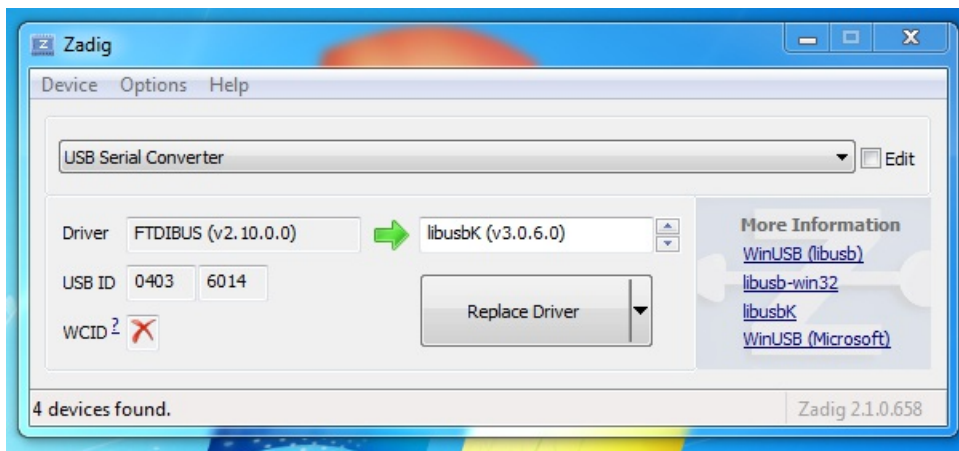


SMT Breakout PCB for SOIC-8, MSOP-8 or TSSOP-8 - 6 Pack!

\$2.95
IN STOCK

ADD TO CART

If you're using Windows, you will need to follow [these instructions \(https://adafru.it/jFx\)](https://adafru.it/jFx) to set up the Zadig tool and replace the default FTDI driver. **Make sure you do this step!**



Download and compile

We have adapted a helper program for FPGAs. [You can check out our version from github \(https://adafru.it/xhE\)](https://adafru.it/xhE).

The easiest way to get started is to just download a zip of the repository:

<https://adafru.it/xhF>

<https://adafru.it/xhF>

Uncompress the zip folder. If you are on mac or linux you will need to install the FTDI library and build the executable. [Check out the FT232H guide on where to download those files. \(https://adafru.it/xia\)](https://adafru.it/xia) For Windows, we provide the .exe already

Wiring

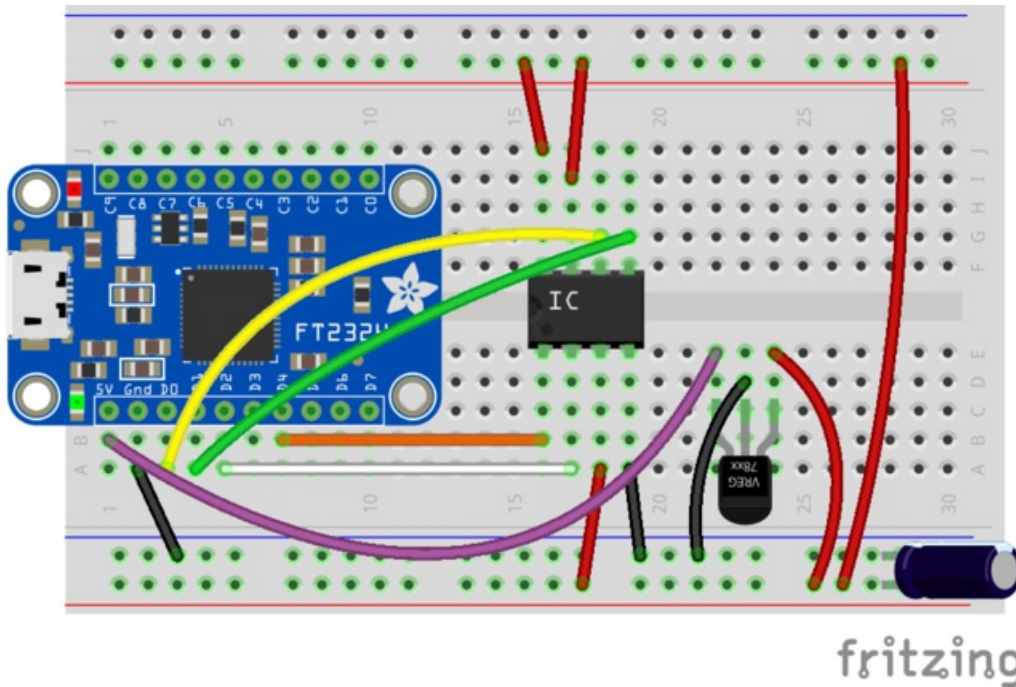
Because SPI flash chips run at 3.3V, you will need to regulate the 5V supply on the FT232H down.

Using the [L4931-3.3 \(https://adafru.it/xhA\)](https://adafru.it/xhA) regulator, make the connections shown in the diagram below.

Make sure to add a **10uF** capacitor from the regulator output to ground for stability!

Wire up your FT232H breakout to your flash chip like this (use a socket or an adapter board if it is not a DIP package)

- D0 on the FT232H to pin 6 on the flash
- D1 on the FT232H to pin 5 on the flash
- D2 on the FT232H to pin 2 on the flash
- D4 on the FT232H to pin 1 on the flash
- pin 3 (Write Protect) on the flash to 3.3V. If you are only reading from the Flash, you can connect this to ground.
- pin 4 (VSS ground) on the flash to ground
- pin 7 (HOLD) on the flash to 3.3V
- pin 8 (VCC power) on the flash to 3.3V



Run the executable from the folder you downloaded. The argument it takes is the path to the binary file you want to write to the flash. An example might look like this

```

C:\msys64\ftdiflash>ftdiflash.exe my_binary_file.bin
init..
reset..
Flash ID: 0x1F 0x84 0x01 0x1F 0x84 0x01 0x1F 0x84 0x01 0x1F 0x84 0x01 0x1F 0x84 0x01 0x1F 0x84 0x01 0x1F 0x84
File size: 512000
Erase 64kB sector at 0x000000..
Erase 64kB sector at 0x010000..
Erase 64kB sector at 0x020000..
Erase 64kB sector at 0x030000..
Erase 64kB sector at 0x040000..
Erase 64kB sector at 0x050000..
Erase 64kB sector at 0x060000..
Erase 64kB sector at 0x070000..
programming..
reading..
VERIFY OK
Bye.
C:\msys64\ftdiflash>

```

That's it!