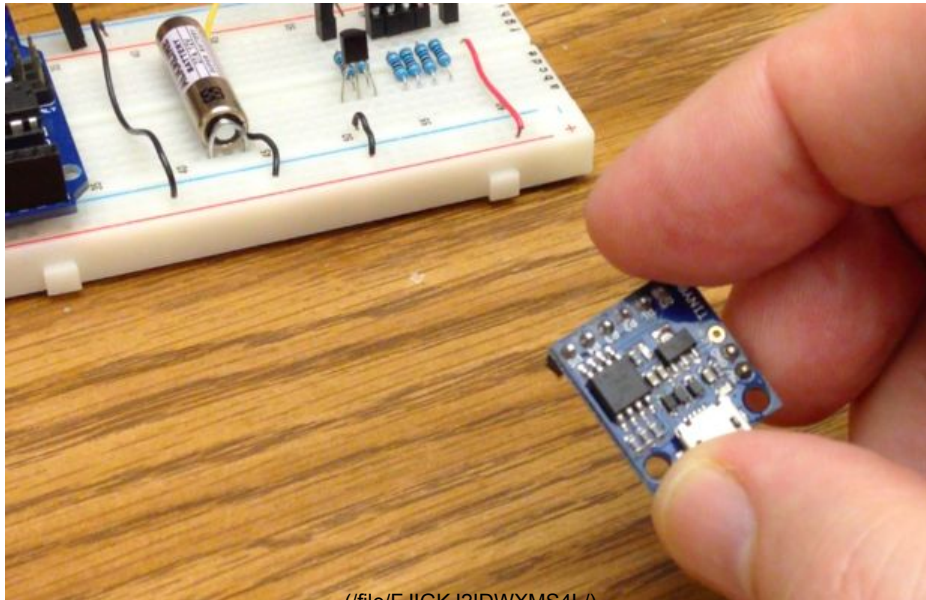


How to Unlock Digispark ATtiny85 and Convert It to a Trinket by

dmjlambert (/member/dmjlambert/) in arduino (/explore/category/technology/keyword/arduino/)

[Download](#)[\(/id/How-to-unlock-Digispark-ATtiny85-and-convert-it-to-\)](#)

8 Steps

[+ Collection](#)[I Made it!](#)[♥ Favorite](#)[Share](#)

About This Instructable

[👁 20,382 views](#)[♥ 52 favorites](#)

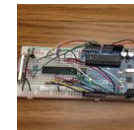
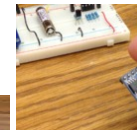
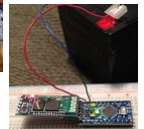
License:

[\(CC\) BY-NC-SA](#)**dmjlambert**
(/member/dmjlambert/)[Follow](#)

32

[\(/member/dmjlambert/\)](#)**Bio:** These Arduinos are driving me up the wall.

More by dmjlambert:

[\(/id/HV-Rescue-Simple/\)](#)[\(/id/How-to-unlock-Digispark-ATtiny85-and-convert-it-to-\)](#)[\(/id/Bluetooth-Serial-Adapter-for-Pro-Mini/\)](#)

Digispark ATtiny85 boards come with the Micronucleus bootloader and with the reset pin disabled. When reset is disabled you get an additional digital pin, and you must upload sketches using the bootloader. You may not use ISP programming.

There are projects where I want to use Digispark and need to enable the reset pin for various reasons:

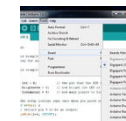
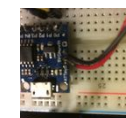
- Reset Digispark via a signal from another device or via a button
- Use ISP programming to upload sketches
- Remove bootloader start up delay
- Upgrade bootloader
- Use a different bootloader, such as Trinket
- Change fuses

This Instructable documents how I unlock the reset pin and program the Digispark with ISP programming to convert it to a Trinket. I have a clone Digispark, which works the same as the Kickstarter Digispark from Digistump.

In this Instructable, I use an Uno as the high voltage programmer, and later I use it as an ISP. You can also use a Nano or Pro Mini, which are simply small Unos.

Using this method you can continue to use the board as a Digispark, or use it as a plain ATtiny85 breakout board and program it via ISP, or use it as a Trinket. If you continue to use the board as a Digispark, the P5 pin has been converted from a regular I/O pin to a reset pin.

Related

[Adding Digispark \(with bootloader\) support to existing Arduino 1.6.x IDE \(/id/Adding-Digispark-with-bootloader-support-to-existing-Arduino-1.6.x-IDE/\)](#)[Refresh your DigiSpark \(clone\) with a smaller bootloader \(/id/Refresh-Your-DigiSpark-clone-With-a-Smaller-Bootloader/\)](#)[Digispark DIY: The smallest USB Arduino \(/id/Digispark-DIY-The-smallest-USB-Arduino/\)](#)[How to Program an ATtiny 85 Digispark \(/id/How-to-Program-an-ATtiny-85-Digispark/\)](#)

Since the Digispark is simply a small break out board for the ATtiny85 MCU, you can also use this Instructable to rescue fuses on a stand alone ATtiny85 MCU.



Turn Your Arduino Into an
ISP //id/Turn Your Arduino

Step 1: Sketch for High Voltage Programming

A sketch for Uno is available here for unlocking ATtiny85:

http://www.rickety.us/wp-content/uploads/2010/03/hv_serial_prog.pde
(http://www.rickety.us/wp-content/uploads/2010/03/hv_serial_prog.pde)

Code is by Paul Willoughby and original article for unlocking bare ATtiny85 ICs here:

<http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/>
(<http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/>)

Sketch:

```
// AVR High-voltage Serial Programmer
// Originally created by Paul Willoughby 03/20/2010
// www.rickety.us slash 2010/03/arduino-avr-high-voltage-serial-programmer/
// Inspired by Jeff Keyzer mightyohm.com
// Serial Programming routines from ATtiny25/45/85 datasheet

// Desired fuse configuration
#define HFUSE 0xDF // Defaults for ATtiny25/45/85
#define LFUSE 0x62

#define RST 13 // Output to level shifter for !RESET from transistor to Pin 1
#define CLKOUT 12 // Connect to Serial Clock Input (SCI) Pin 2
#define DATAIN 11 // Connect to Serial Data Output (SDO) Pin 7
#define INSTOUT 10 // Connect to Serial Instruction Input (SII) Pin 6
#define DATAOUT 9 // Connect to Serial Data Input (SDI) Pin 5
#define VCC 8 // Connect to VCC Pin 8

int inByte = 0; // incoming serial byte Computer
int inData = 0; // incoming serial byte AVR

void setup()
{
  // Set up control lines for HV parallel programming
  pinMode(VCC, OUTPUT);
  pinMode(RST, OUTPUT);
  pinMode(DATAOUT, OUTPUT);
```



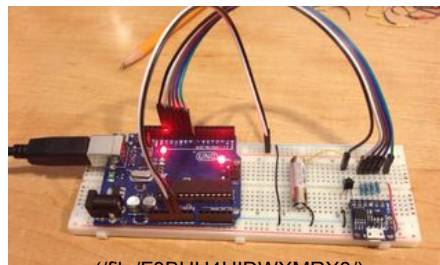
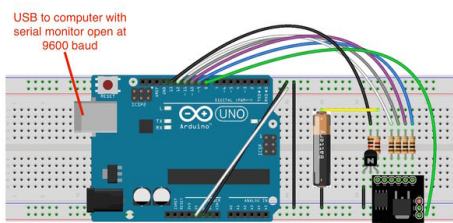
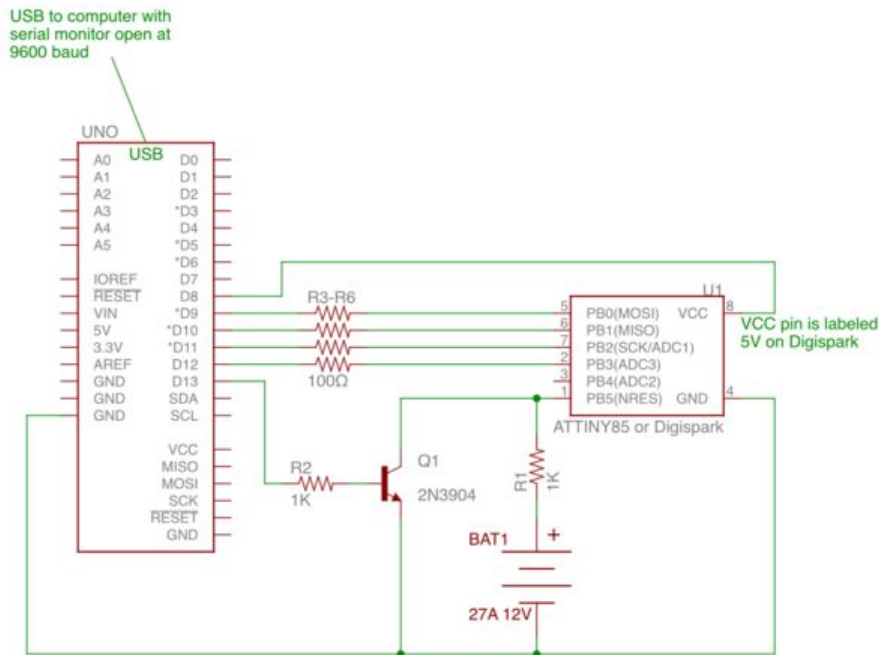
HV Programmer for

Download (<https://cdn.instructables.com/ORIG/FEG/RV2/IDWXMWKM/FEGRYV2IDWXMWKM.zip>)

(<https://cdn.instructables.com/ORIG/FEG/RV2/IDWXMWKM/FEGRYV2IDWXMWKM.zip>)

advertisement

Step 2: Wiring



I use an Uno and a breadboard with a 2N3904 transistor, 2 ea 1K resistors for the connections to the transistor, and 4 ea 100 ohm resistors for the programming connections between the Uno and Digispark. You may be able to use 1K resistors for the programming connections on a bare ATTtiny85, but I found the assembled Digispark needs a lower value. It is probably because I need to overpower the onboard pull-up resistors during programming.

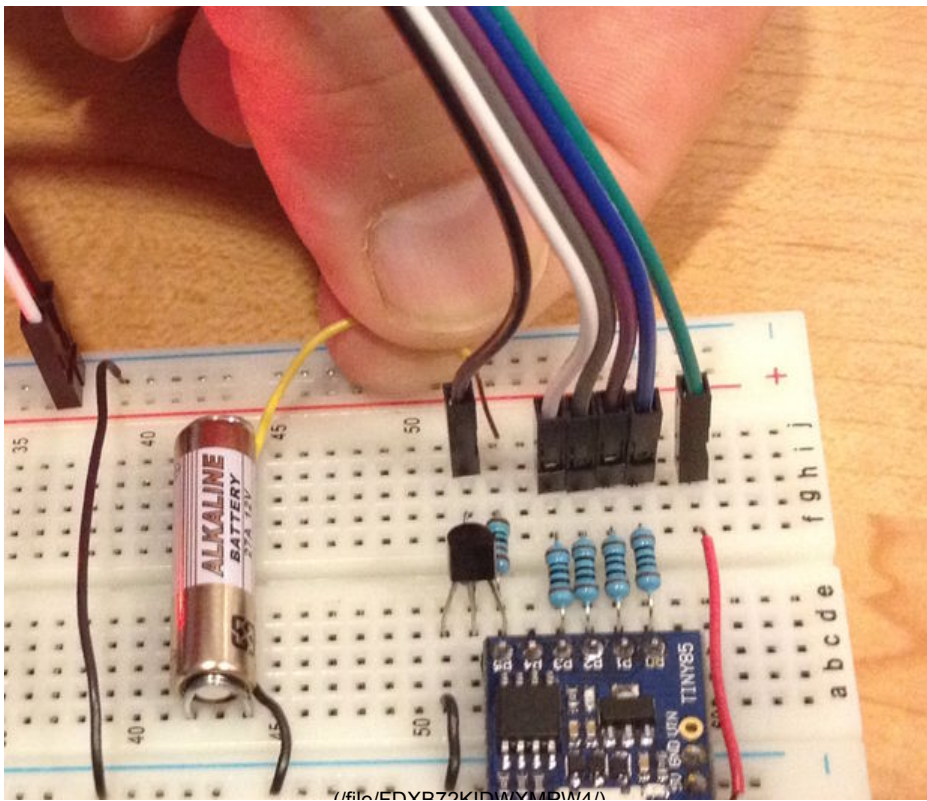
I use pieces of a paperclip for a battery holder and bend them to get a little spring tension.

I like to add all header pins to my Digispark, except VIN, so it fits on a breadboard and connects to the power rails and terminal strips.

Leave the 27A battery disconnected at first, and connect it after the Uno with HV programming sketch is attached to the computer via USB and powered up. This is because if the battery is connected before power is applied to the Uno, the digital pin 13 on the Uno will not have a high signal and switch the transistor, and the battery would then immediately begin sending 12V to the Digispark's pin 5.

When done programming, disconnect the battery first, then unplug the USB cable from the Uno.

Step 3: Running the HV Programming Sketch



When you connect the Uno via USB and connect the battery, open the Serial Monitor window. Set the Line Ending to Both NL & CR and baud rate to 9600. You will see the text "Enter a character to continue" scrolling through the window. On the send line type a character, such as "a", and click Send. You will then see the programmer go to work, and see some results:

```
Enter a character to continue
Enter a character to continue
Enter a character to continue
Enter a character to continue
Enter a character to continue
Enter a character to continue
97
Entering programming Mode

lfuse reads as FF
hfuse reads as FF
efuse reads as FF

Writing hfuse
Writing lfuse

lfuse reads as FF
hfuse reads as FF
efuse reads as FF

Exiting programming Mode

13
Entering programming Mode

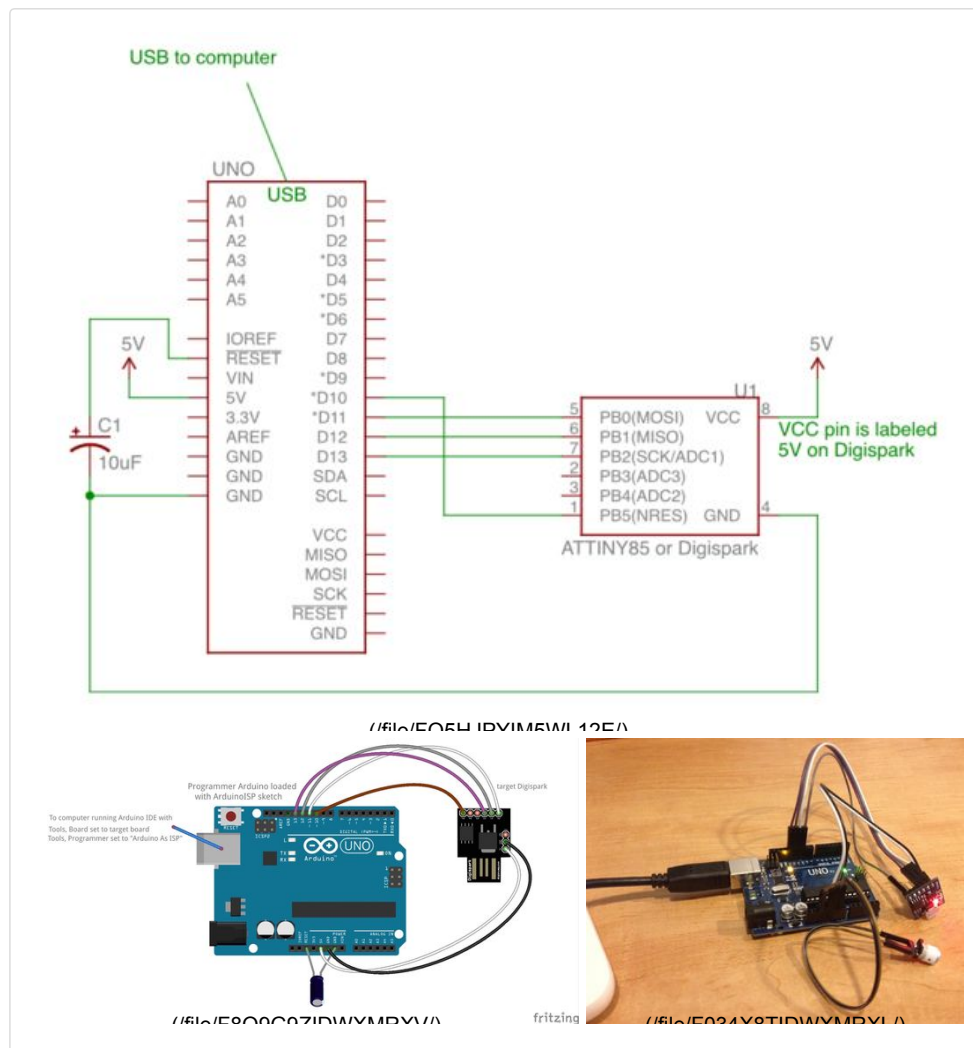
lfuse reads as F1
hfuse reads as 55
efuse reads as FE
```

When it says the final "Exiting programming Mode" disconnect the battery, disconnect the Uno, and disconnect the Digispark from the breadboard.

Digispark reset fuse unlock with high voltage programming



Step 4: Programming Digispark Via ISP



You can now use ISP programming to program the ATtiny85 on the Digispark as you see fit. I use the Uno loaded with the example ArduinoISP sketch to do that.

You can continue to use the Micronucleus bootloader and the Digispark-specific version of Arduino IDE. Pin 5 is now a reset pin instead of a digital pin, so make adjustments in your sketches for that.

OR

You can install an ATtiny85 core in your Arduino IDE and program it using ISP just like a stand alone ATtiny85. The Digispark has been converted to an ATtiny85 breakout board.

OR

You can install the Trinket bootloader on it and use it as a Trinket. See Adafruit's website and tutorials on the Trinket.

Command line for avrdude

Your command line for avrdude may use different directory paths and different ports. To learn what the correct command line looks like for your computer, I think it is easiest to upload the ArduinoISP sample sketch to the Uno, set Tools, Programmer to Arduino as ISP, and use the Uno to burn the bootloader on another Arduino of yours, such as a Nano, Pro Mini. You can actually leave the Uno disconnected and attempt to burn a bootloader to an imaginary device if you want. If you have gone into the Arduino IDE preferences and checked show verbose output during upload, the command line for avrdude which attempted to burn the bootloader will be shown to you. All you need to do is change the last part of the line that gives the path to the bootloader, and replace it with the path to your bootloader, and add the fuse settings. And you may need to also change the -p option, which is where you specify the processor type.

Continue to use board as a Digispark and upgrade Digispark Micronucleus bootloader

I downloaded the latest Micronucleus bootloader from here:

<https://github.com/micronucleus/micronucleus/tree...>

(<https://github.com/micronucleus/micronucleus/tree/master/firmware/releases>)

And loaded it onto the Digispark like this:

```
/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -C/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/cu.usbmodem411 -b19200 -Uflash:w:t85_default_micronucleus.hex -U lfuse:w:0xF1:m -U hfuse:w:0xD5:m -U efuse:w:0xFE:m
```

Use board as a Trinket

I downloaded the Trinket bootloader from here:

<https://github.com/adafruit/Adafruit-Trinket-Gemma...>

(<https://github.com/adafruit/Adafruit-Trinket-Gemma-Bootloader>)

And loaded it like this:

```
/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -C/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/cu.usbmodem411 -b19200 -Uflash:w:trinketgemma_v1.hex -U lfuse:w:0xF1:m -U hfuse:w:0xD5:m -U efuse:w:0xFE:m
```

If it has the Trinket bootloader on it, it is now a Trinket. If you are new to Trinket, look at the Adafruit website and tutorials about Trinket to find out how to set it up your IDE for programming it. It is a little different than other Arduinos in that you do not need to select a port, but instead you set Programmer to USBtinyISP. Just like with Digispark, the Trinket does not use Serial Monitor the same way you might do with other Arduinos.



bootloaders.zip

Download (<https://cdn.instructables.com/ORIG/F0U/5BJJ/IDWXN4KM/F0U5BJJIDWXN4KM.zip>)

(<https://cdn.instructables.com/ORIG/F0U/5BJJ/IDWXN4KM/F0U5BJJIDWXN4KM.zip>)

Step 5: Reset Disable Fuse

If I want to disable the reset pin again and get that additional digital pin 5, I would replace the

```
hfuse:w:0xD5:m
```

with

```
hfuse:w:0x55:m
```

Of course, that means I would need to use the high voltage programmer to unlock it again if I want to use ISP programming with it again. I use this command line to just set that fuse to disable the reset pin:

```
/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -C/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/cu.usbmodem411 -b19200 -U hfuse:w:0x55:m
```

So, after I load a sketch via ISP that needs to use digital pin 5, I can set the high fuse to 0x55 to start using that pin and to lock the sketch in. Or, I can use that extra pin when I program it with the Trinket or Micronucleus bootloader. The Digispark and the Trinket both have equal ability to use pin 5 as a reset pin, or as a data pin, depending on the setting of the fuse. There is nothing special about the design of the Digispark to give it the extra data pin, it is simply a matter of deciding how you want to set the fuse. You can easily set the fuse either way depending on what you need for the project you are working on.

Step 6: Fuses

In the links section I give the link to a fuse calculator, which shows what the various fuses do. My preferred fuses for both Trinket and Digispark:

- L:F1
- H:D5
- E:FE
- H:55 for reset disable

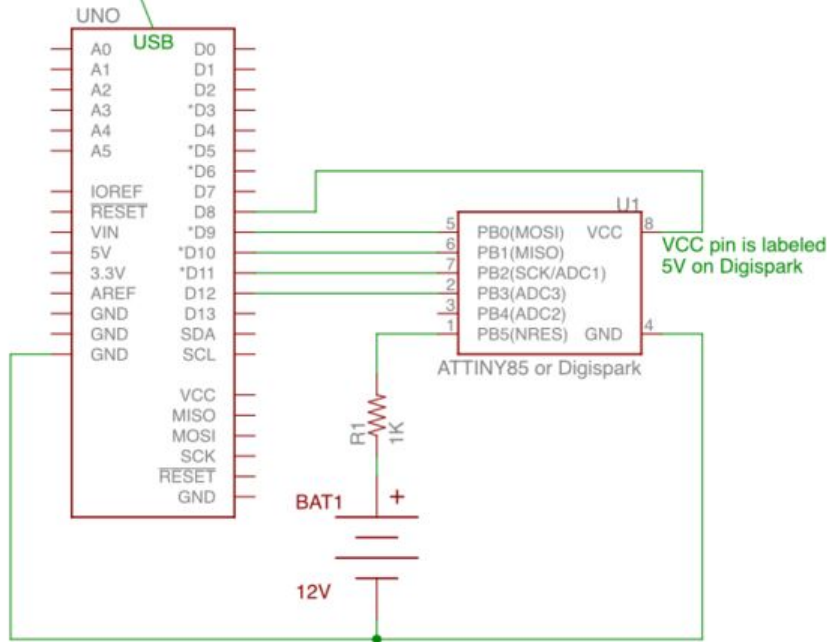
There is a minor bug in avrdude. At the end of programming when it shows you the fuses it has set, it looks like this:

```
avrdude: safemode: lfuse reads as F1
avrdude: safemode: hfuse reads as D5
avrdude: safemode: efuse reads as FE
avrdude: safemode: Fuses OK (H:FE, E:D5, L:F1)
```

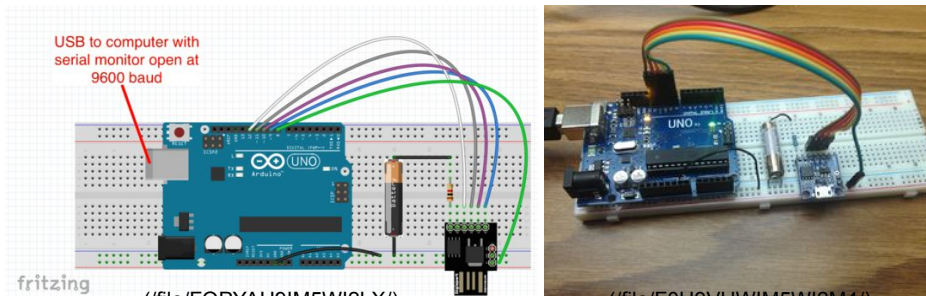
Note the summary line for Fuses OK shows H and E fuse values reversed.

Step 7: Transistors? We Don't Need No Stinking Transistors!

USB to computer with
serial monitor open at
9600 baud



U1: ATTINY85 or Digispark



Here is the alternate wiring

Ivoh's comment reminded me about the way I provided high voltage to program EPROMs back in the day. The data sheet for the ATtiny85 talks about the timing and sequence between applying 5V to VCC and 12V to RESET, but that timing and sequence does not appear to be critical. Alternate method for the high voltage programming step:

Instead of switching the 12V that is going to the reset pin, just connect it by hand. Connect the 12V first, before you power VCC (you will send power to VCC by running the sketch, which powers up the Digispark or ATtiny85 from the Arduino's pin 8). It may take a couple of tries, because sometimes nothing happens after you see the message "Entering programming Mode" in the serial monitor. So, if you need to start over, just disconnect the USB cable from the Arduino and remove the 12V battery and close the serial monitor. Then connect the battery, followed by connecting the USB cable, then open serial monitor and when prompted, enter a character. The programming happens quickly, just as shown in the video of Step 3. Proceed to Step 4 for ISP programming.

Step 8: Links

HV programming info

<https://arduino diy.wordpress.com/2015/05/16/high-voltage-programmingunbricking-for-attiny/>
(<https://arduino diy.wordpress.com/2015/05/16/high-voltage-programmingunbricking-for-attiny/>)

<http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/>
(<http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/>)

You can also use this sketch/schematic for Digispark

<https://www.instructables.com/id/HV-Rescue-Simple/>
(<https://www.instructables.com/id/HV-Rescue-Simple/>)

ATtiny85 fuse info

<http://eleccelerator.com/fusecalc/fusecalc.php?chip=attiny85&LOW=F1&HIGH=55&EXTENDED=FE&LOCKBIT=FF>
(<http://eleccelerator.com/fusecalc/fusecalc.php?chip=attiny85&LOW=F1&HIGH=55&EXTENDED=FE&LOCKBIT=FF>)

<http://www.atmel.com/devices/attiny85.aspx>
(<http://www.atmel.com/devices/attiny85.aspx>)

Tiny core

<https://github.com/SpenceKonde/ATTinyCore>
(<https://github.com/SpenceKonde/ATTinyCore>)

Digispark info

<http://digistump.com/wiki/digispark> (<http://digistump.com/wiki/digispark>)

<http://digistump.com/board/index.php/topic,1039.0.html>
(<http://digistump.com/board/index.php/topic,1039.0.html>)

<https://s3.amazonaws.com/digispark/DigisparkSchematicFinal.pdf>
(<https://s3.amazonaws.com/digispark/DigisparkSchematicFinal.pdf>) (schematic of original USB version)

<https://www.instructables.com/id/Adding-Digispark-with-bootloader-support-to-existi/> (<https://www.instructables.com/id/Adding-Digispark-with-bootloader-support-to-existi/>) (note: outdated as of Nov 2015 since Digispark has now added Boards Manager support)

<https://www.instructables.com/id/Digispark-DIY-The-smallest-USB-Arduino/step2/Burning-bootloader-to-ATTINY85/>
(<https://www.instructables.com/id/Digispark-DIY-The-smallest-USB-Arduino/step2/Burning-bootloader-to-ATTINY85/>)

<https://github.com/micronucleus/micronucleus/tree/master/firmware/releases>
(<https://github.com/micronucleus/micronucleus/tree/master/firmware/releases>)

<http://chioszrobots.com/2015/02/12/digispark-kickstarter-usb-development-board-attiny85/> (<http://chioszrobots.com/2015/02/12/digispark-kickstarter-usb-development-board-attiny85/>) (schematic of micro USB version)

Trinket info

<https://learn.adafruit.com/introducing-trinket/introduction>
(<https://learn.adafruit.com/introducing-trinket/introduction>)

<https://github.com/adafruit/Adafruit-Trinket-Gemma-Bootloader>
(<https://github.com/adafruit/Adafruit-Trinket-Gemma-Bootloader>)

Exchangeable soul, the inverted prequel


<http://hackaday.com/2014/01/22/body-of-a-trinket-soul-of-a-digispark/>
(<http://hackaday.com/2014/01/22/body-of-a-trinket-soul-of-a-digispark/>)


<https://www.hackster.io/rayburne/trinket-firmware-exorcism>
(<https://www.hackster.io/rayburne/trinket-firmware-exorcism>)


Electronic Edification


<http://www.ermicro.com/blog/?p=423> (<http://www.ermicro.com/blog/?p=423>)

Comments



 We have a be nice comment policy.
Please be positive and constructive.

 I Made it!

 Add Images

Post Comment



ambimod.by (/member/ambimod.by/)

2017-06-07

Reply

Hi,

I really want to thank you for the information found here. I spent a lot of time trying to change the code of a sketch for a trinket to use it with a Digispark but with no success.

I took the initiative to write my own tutorial in french based on yours and sharing it for readers coming on my blog.

I also use your pictures putting mine beside it to show the connexion that i made with my arduino nano.

If you want to take a look on :

<https://ambimod.jimdo.com/2017/06/08/tuto-comment-...>

(<https://ambimod.jimdo.com/2017/06/08/tuto-comment-convertir-un-digispark-%C3%A0-1-en-un-trinket-%C3%A0-7/>)

Thank you again, you save my life /;-)



CaptClaude (/member/CaptClaude/)

2016-02-11

Reply

When I found the Digispark clones on Banggood, I bought a handful because I became addicted to SMALL -- if all you want to do is drive a couple of LEDs or some NeoPixels, who needs an Uno, right? Cheap too, but mostly small. This is a great, detailed tutorial with sharp pictures, details and (forgive me for this) understandable English. Very educational, I will benefit from your experience, thanks. You are bookmarked.



CaptClaude (/member/CaptClaude/) ▶ CaptClaude (/member/CaptClaude/)

Reply

Not sure why that comment posted twice and even more not sure why it took me so long to see that it had posted twice and delete the second one.

2017-02-20



ErikA45 (/member/ErikA45/) made it!

2017-02-04

Reply

I made an all-in-one plug-in board with HV programmer for Digispark (left) and regular ICSP programmer for Digispark (right) and ATmega328/168. I have room for a 5V-to-12V booster lower left to be added (item ordered and in transit). The capacitor for the reset is activated by a simple jumper. I added two tiny LED (from analog outputs 18 & 19): one that flashes at low intensity while waiting for character entry, and runs full light when in program mode, and one that runs full light when 12V is active. ..a bit of gadgetry but fun to see.

Delay time for 12V kick-in in program adapted to 40usec. Fuses for Digispark adapted equally: see my previous note here.

(<https://cdn.instructables.com/F1P/KMR9/IYOQ91TL/F1PKMR9IYOQ91TL.LARGE.jpg>)

ErikA45 (/member/ErikA45/) made it! ▶ ErikA45 (/member/ErikA45/)

Reply

Forgot to add this spaghetti-picture-underside.

2017-02-04

(<https://cdn.instructables.com/FYH/S6HO/IYOQ93IY/FYHS6HOIYOQ93IY.LARGE.jpg>)

ErikA45 (/member/ErikA45/)

2017-02-01

Reply

How do you set the extended fuse with this program? Or don't you need a HV programmer for this fuse?

dmjlambert (/member/dmjlambert/) ▶ ErikA45 (/member/ErikA45/)

Reply

I think that is the case, you don't need a HV programmer for the extended fuse because nothing about the extended fuse could cause the chip to be non-programmable via ISP. I did not study this programmer code in detail or alter it, it is from a different blog and I just adapted the schematic a little bit to work with Digispark boards.

2017-02-01

I have another Instructable <https://www.instructables.com/id/HV-Rescue-Simple/> that is based on Jeff Keyzer's version 2 of his HV Rescue Shield. I simplified the hardware to make it breadboard-based with fewer components, and enhanced the sketch to work with most Arduinos and give it a menu with control of the fuse values.

ErikA45 (/member/ErikA45/) ▶ dmjlambert (/member/dmjlambert/)

Reply

I looked at this instruction; can you upload the "HV Rescue Simple schematic.png" with higher resolution? The current one is barely readable. Thks!

2017-02-01

dmjlambert (/member/dmjlambert/) made it! ▶ ErikA45 (/member/ErikA45/)

I just downloaded the zip file and in it is the schematic. 2017-02-01 Reply
I was able to zoom the schematic to any size with my favorite image editing program. I think you just need to zoom in if it is too small.

(<https://cdn.instructables.com/FE3/4IF6/IYNLJRE2/FE34IF6IYNLJRE2.LARGE.jpg>)

ErikA45 (/member/ErikA45/) 2017-01-18 Reply
Fuse settings are wrong if you want to emulate the Digispark settings without pin5 reset disabled. Lfuse needs to be E1 and Hfuse need to be DD. Efuse should be FE.

dmjlambert (/member/dmjlambert/) ▶ ErikA45 (/member/ErikA45/) 2017-01-18 Reply
I don't think I agree with using the word "wrong." Those are my preferred fuse values. The differences between those and what the Digispark comes set with are brown out detection, EESAVE, and start up time. Digispark's official values are on this page:
<http://digistump.com/wiki/digispark/tutorials/programming>

ErikA45 (/member/ErikA45/) ▶ dmjlambert (/member/dmjlambert/) 2017-01-18 Reply
You are right, your instructable says "...return to a trinket.", not "return to Digispark". I should have used the word "different".
However, it would still be usefull for anyone wanting to use their Digispark with the USB downloader, that the fuse settings must be adapted in your program.

ErikA45 (/member/ErikA45/) 2017-01-18 Reply
Works well, except on two Digispark where I had the RSTDSBL set (pin5 as output, no reset). On one Digispark the program hangs 2 minutes at reading lfuse, then finalises ok. The other Digispark causes the program to hang after lfuse reading as E0.
Strange thing is it works well on Digisparks where RSTDSBL is not set.
What could be the reason?
Erik

ErikA45 (/member/ErikA45/) ▶ ErikA45 (/member/ErikA45/) 2017-01-18 Reply
I think I may have solved it: I increased "delayMicroseconds(20);" to 60: page 156 of the datasheets specifies between 20 and 60 microseconds delay between Vcc and Reset High..

zonkerharris (/member/zonkerharris/)

2017-01-16

Reply

This is an interesting instructable, in regards to the reference to the 12v battery . I haven't seen a battery on other programmers, so this bears making time for more reading. Thanks for the clarification on why you used 100-ohm resistors, versus the 1k used in the other page you referenced. :-)

TryxCorp (/member/TryxCorp/)

2016-12-25

Reply

Great instructable and thank you for sharing it :-D

Your instructable solved my problem to an extent.

When I tried to program digispark's attiny85 to run a blink code with a delay of 1000ms, the delay doesnt seem to be 1000ms. Instead it is observed to be around 16000ms. I measured it using a stopwatch I programmed the attiny85 with the method that you have decribed. Any solutions to the above problem? TIA :-)

zonkerharris (/member/zonkerharris/) ▶ **TryxCorp (/member/TryxCorp/)**

Reply

2017-01-16

You may also want to check out this Fuse Calculator (<http://www.engbedded.com/fusecalc>), if you want to get a bit deeper into the AVR programming side. Clock speed is just one facet you can modify in that programming sketch. This helped me to get the Tiny85 running at 16 MHz internal clock. More clues can be found in one of my Instructables, <https://www.instructables.com/id/IKEA-Star-With-ATtiny-and-NeoPixels/>

dmjlambert (/member/dmjlambert/) ▶ **TryxCorp (/member/TryxCorp/)**

Reply

2016-12-25

I think the fuse settings and the core you use when you compile a sketch for it determine how fast the processor runs. It sounds like your sketch is compiled for 16MHz but your processor is actually running at 1MHz if you are noticing that speed difference. If you have put the Micronucleus bootloader back onto it and are programming it as a Digispark, then you should be able to find information on the Digispark website or wiki about speeds. If you have put the Trinket bootloader on it and are programming it as a Trinket, you should be able to find information in Adafruit's tutorials or forums about that. There are a variety of cores you can use to add on to the Arduino IDE to program plain ATtiny85's using an ISP programmer, and most of those will involve using the Burn Bootloader function of the IDE to set the fuses to the appropriate values. Help for using those cores could probably be found at the arduino.cc website in the Microcontrollers section of the forum.

ntewinkel (/member/ntewinkel/)

2016-06-21

Reply

Thanks for this great Instructable!

This worked really nicely. I had a bricked clone-spark due to a failed bootloader attempt. This brought it back to life and upgraded it with the addition of the Digispark bootloader (so it's much easier to program now).

Worked like a charm - it was so easy I almost thought I'd screwed it up :D

In case it helps others to know: I used a PN2222A transistor, and a direct-clip-on adapter that allowed me to directly connect to the chip (I haven't put headers on the board, to keep it tiny).

ivoh (/member/ivoh/)

2015-11-29

Reply

Thanks! I was messing with my digisparks bootloader and bricked it, this tutorial helped me save it! After hours of searching on the web, this is the absolute best tutorial i found! It should be noted that I did not use the 100ohm resistors or pin 13 to transistor and it still worked! (although it was probably risky), i just connected the 1kohm resistor from the battery to pin 5 manually without a transistor. Thanks again!

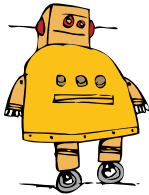
dmjlambert (/member/dmjlambert/) ▶ ivoh (/member/ivoh/)

2016-03-24

Reply

Thanks, I have edited the Instructable to include the alternate wiring.

↓ More Comments



Newsletter

Let your inbox help you discover our best projects, classes, and contests. Instructables will help you learn how to make anything!

About Us

[Who We Are \(/about/\)](/about/)

[Advertise \(/advertise/\)](/advertise/)

[Contact \(/about/contact.jsp\)](/about/contact.jsp)

[Jobs \(/community/Positions-available-at-Instructables/\)](/community/Positions-available-at-Instructables/)

[Help \(/id/how-to-write-a-great-instructable/\)](/id/how-to-write-a-great-instructable/)

Find Us

[Facebook \(http://www.facebook.com/instructables\)](http://www.facebook.com/instructables)

[Youtube \(http://www.youtube.com/user/instructablestv\)](http://www.youtube.com/user/instructablestv)

[Twitter \(http://www.twitter.com/instructables\)](http://www.twitter.com/instructables)

[Pinterest \(http://www.pinterest.com/instructables\)](http://www.pinterest.com/instructables)

[Google+ \(https://plus.google.com/+instructables\)](https://plus.google.com/+instructables)

Resources

[For Teachers \(/teachers/\)](/teachers/)

[Residency Program \(/pier9residency\)](/pier9residency)

[Gift Premium Account \(/account/give?source=footer\)](/account/give?source=footer)

[Forums \(/community/?categoryGroup=all&category=all\)](/community/?categoryGroup=all&category=all)

[Answers \(/tag/type-question/?sort=RECENT\)](/tag/type-question/?sort=RECENT)

[Sitemap \(/sitemap/\)](/sitemap/)

[Terms of Service \(http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721\)](http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721) |

[Privacy Statement \(http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079\)](http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079) |

[Legal Notices & Trademarks \(http://usa.autodesk.com/legal-notices-trademarks/\)](http://usa.autodesk.com/legal-notices-trademarks/) | [Mobile Site \(https://www.instructables.com\)](https://www.instructables.com)



<http://usa.autodesk.com/adsk/servlet/pc/index?id=20781545&siteID=123112>

© 2017 Autodesk, Inc.