USB SNES Controller

by **timeblade0** on November 4, 2009

Table of Contents

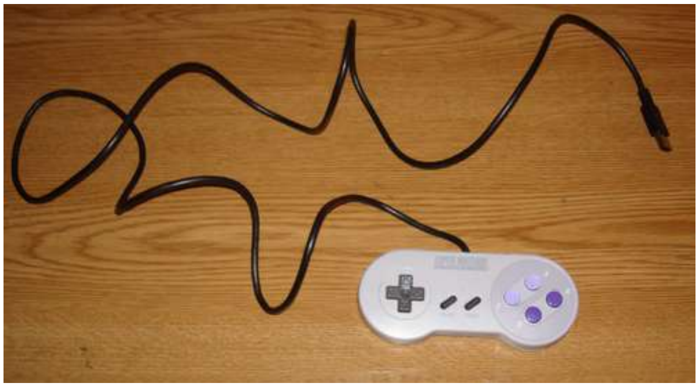
Intro: USB SNES Controller

First instructable. Let me know what it needs and fixes.

I did not make the circuit or program. I am just making a guide that shows step by step assembly of how to make the USB SNES controller. The original page is here: www.raphnet.net/electronique/snes_nes_usb/index_en.php

This guide will convert a regular snes controller into a usb gamepad. It is tested working on XP, Vista, 7, and PS3. It should work on mac and linux but I have not tested them. The device is recognized as a standard HID device.

-You do not need to program anything for this guide. The program is already compiled and ready to be flashed to your chip.
-You do need basic soldering skills.
-Basic schematic reading skills helps, but I will try to explain every step.



Step 1: Parts Needed

Parts List:

-Atmega8 DIP microcontroller
-AVR programmer (usb, parallell or serial. I use usb.)
-12MHz Crystal osscilator
-[2] 3.6 volt zener diode
-1.5K Resistor (anywhere from 1.3K to 1.7K should work)
-[2] 68ohm resistor (anywhere from 60 to 75 should work)
-USB male cable
-SNES controller (first party works better. wires colors match guide.)
-Small wire.
-PCB - www.radioshack.com/product/index.jsp - $2
-Breadboard. This makes it much easier to test before putting into a controller.
www.radioshack.com/product/index.jsp - $9

Tools:

-Wire Stripper
-Wire Cutter
-Soldering Iron
-Solder
-Needle nose pliers (makes it alot easier to place and move wires on breadboard)
-Dremmel or cutting tool. (chip will not fit into unmodified controller.

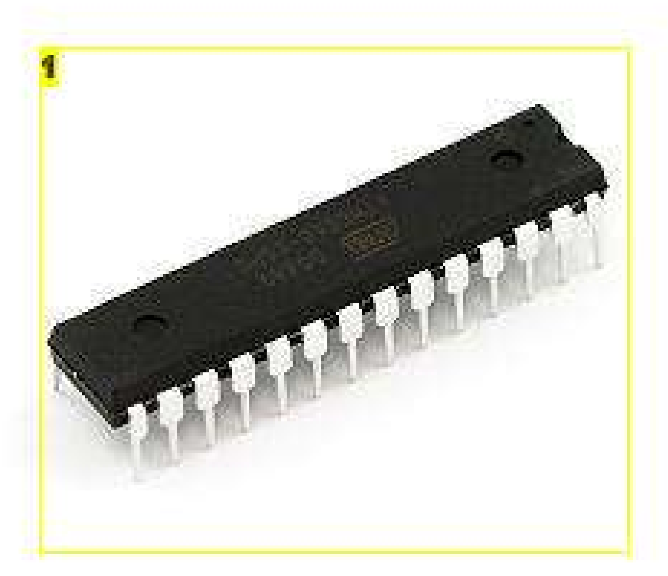


Image Notes
1. Atmega8 Microcontroller

Image Notes
1. USB Programmer. $20 on ebay. This is the one I use.

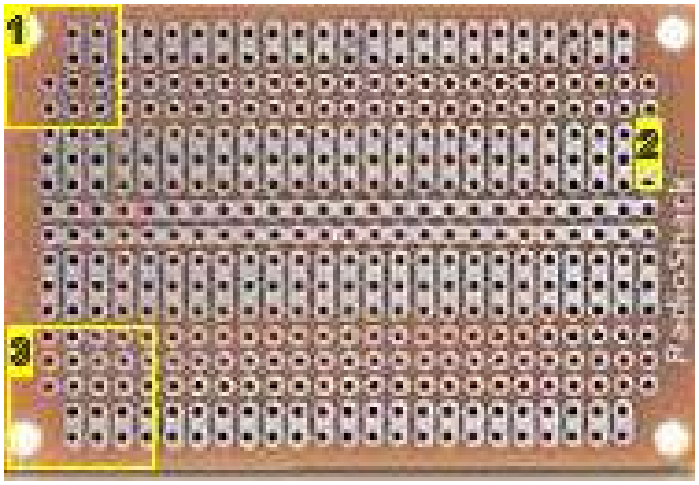


Image Notes
1. PCB. This board can be found at Radioshack for $2. It needs to be cut down to fit in the controller.
2. This pin can have 2 wires connected to it.
3. The atmega8 chip should not be on this side.

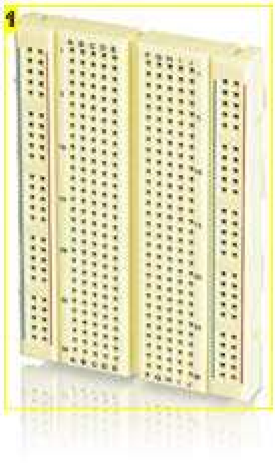


Image Notes
1. A small board is all you need. A bigger one will work, but it costs more and takes up alot more desk space.

Image Notes
1. 12MHz Crystal

Step 2: Prepare Wires

-Take a usb cable and cut off the non male end. You only need the cable and male usb.
-Strip the USB cable and all 4 connections inside. (some usb cables have 5 wires. This guide does not need the 5th wire, so it can be cut off.

-Open the SNES controller. Cut the connection wires close to the board, but leave enough wire so they can be solderd to a pcb. Better to leave too much wire and cut it off if you don't need it.

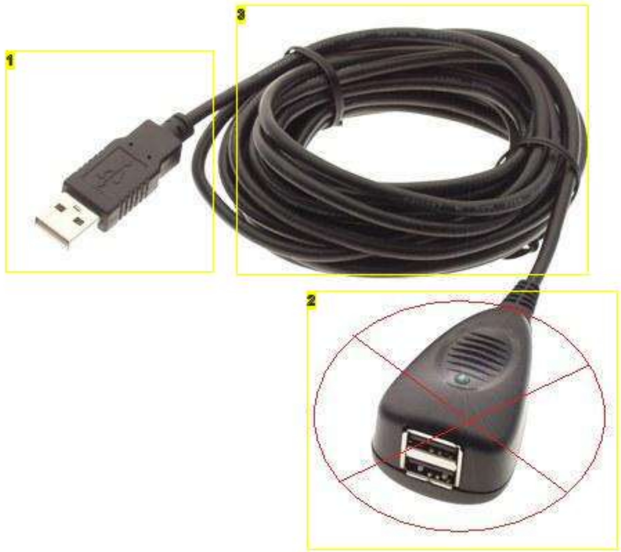


Image Notes
1. Keep this connection.
2. Cut this off.
3. Might not need this much wire, but a long controller cord is nice.

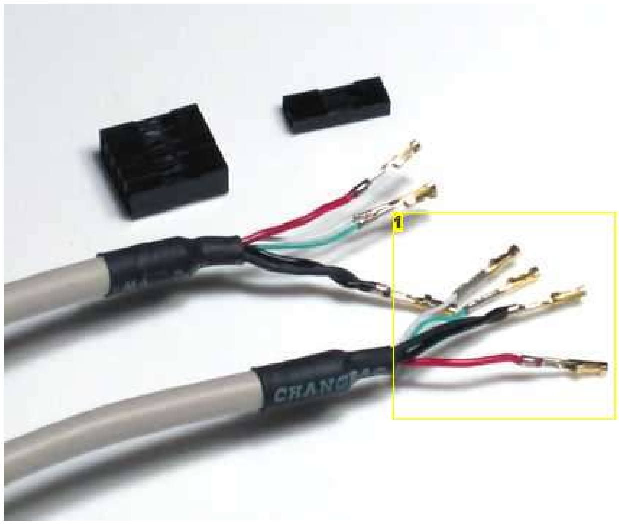


Image Notes
1. 5th wire is usually Brown or Gray. Cut off any color other than red, black, white, and green.

Image Notes
1. Unmodified SNES controller inside.

Step 3: Programming the Atmega8

I used a usb programmer for my design. Serial and parallel will work to, but the programmer software will be a little different from mine.

The program can be downloaded here: (right click save target as...)
www.raphnet.net/electronique/snes_nes_usb/releases/nes_snes_db9_usb-1.7.hex

All atmega chips need fuse bytes. Don't worry about what they do, just know that this design needs:
high byte = **0xc9**
low byte = **0x9f**

Your program for downloading to the chip should have an option to set these fuses.

When you have it programmed, you are ready to start bread boarding.

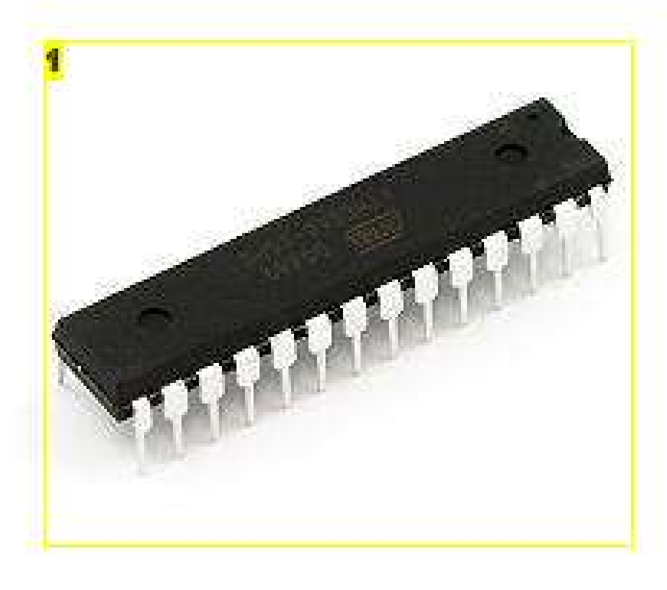


Image Notes
1. Atmega8 Microcontroller

Image Notes
1. USB Programmer. $20 on ebay. This is the one I use.

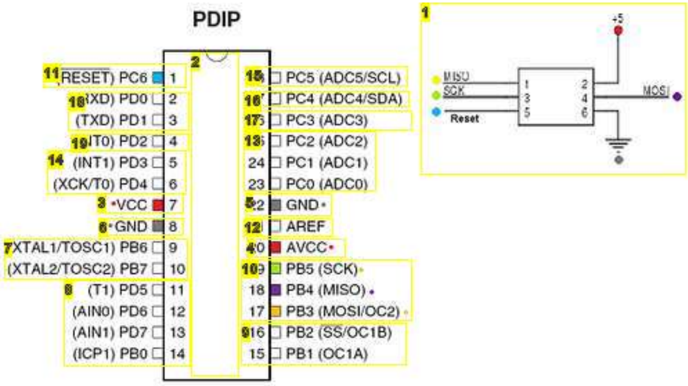


Image Notes
1. Your programmer will have these connections. Each programmer is different, so you will have to find a datasheet for you programmer to find the pins.
2. Atmega8 Chip Note the curve at the top. If you put the chip in backwards it will not work.
3. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
4. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
5. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
6. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
7. 12MHz Oscillator goes here. It can be connected either way.
8. Unused pins. I cut them off with a dremmel to save space.
9. More unused. Can be cut off.
10. Connect to programmer. Can be cut off after programming and testing.
11. Connect to Programmer. Can be cut off after programming and testing.
12. Unused Pin
13. Unused pins. Keep or cut off. Your choice.
14. Unused pins. Keep or cut off.
15. SNES Clock
16. SNES Latch
17. SNES Data
18. USB Data -. Also Zener Diodes and Resistors. See full schematic.
19. USB Data +. Also Zener Diodes and Resistors. See full schematic.

Step 4: Breadboarding

I will assume you know the basics of using a breadboard. If not, there are many guides for that.
I will also assume that you know how to solder wires.

If you can not read the picture, go here:
www.raphnet.net/electronique/snes_nes_usb/sch-revD.png

Start by solding each of the 4 usb wires and each of the 5 snes wires. Solder the ends so that they can be plugged into the breadboard.

Attach the programmed microcontroller and start to wire the components.

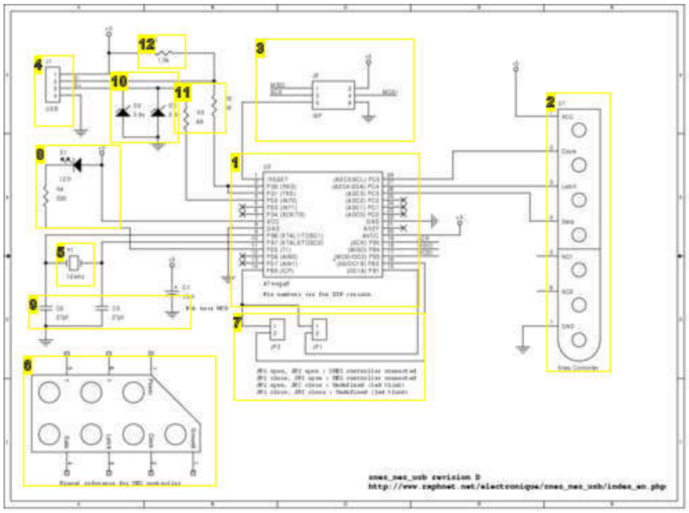

1
2
3
4
5
6
7
8
9
10
11
12
snes_nes_usb revision D
http://www.raphnet.net/electronique/snes_nes_usb/index_en.php

Image Notes
1. Atmega8 chip.

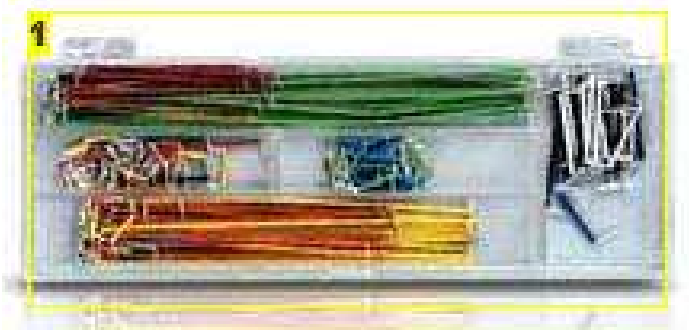


Image Notes
1. Very useful breadboard wiring kit from Radioshack.

2. SNES controller port (on controller view)
3. This is the programmer. It is not needed when fully assembled.
4. USB male connection
5. Timing Crystal
6. NES Controller port. The NES controller has the same wires as an SNES contoller. If you wire the connections right, this same guide will work for an NES controller. I will do a seperate guide for NES to USB later.
7. Open + Open = SNES This part can be omitted for the SNES version.
8. Power LED indicator. I did not include it.
9. Technically, capacitors are not needed. They are a good engineering practice to include becuase they filter the power. I did not use them.
10. 2 3.6V zener diods
11. 68 ohm resistors
12. 1.5K resistors

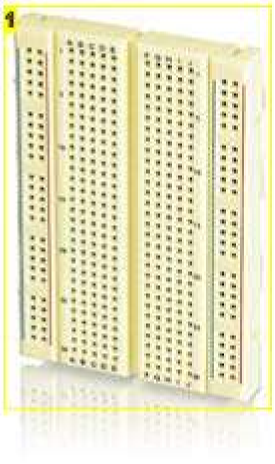


Image Notes
1. A small board is all you need. A bigger one will work, but it costs more and takes up alot more desk space.

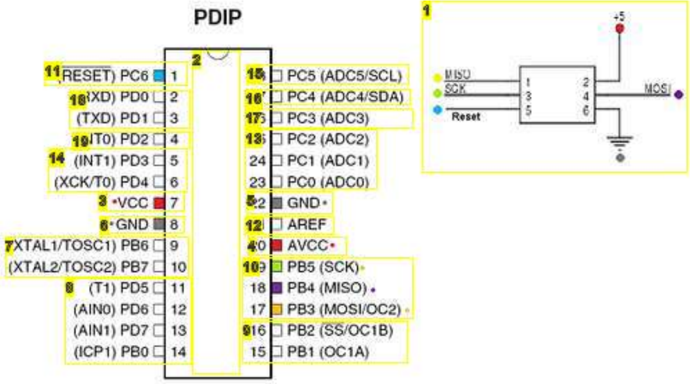
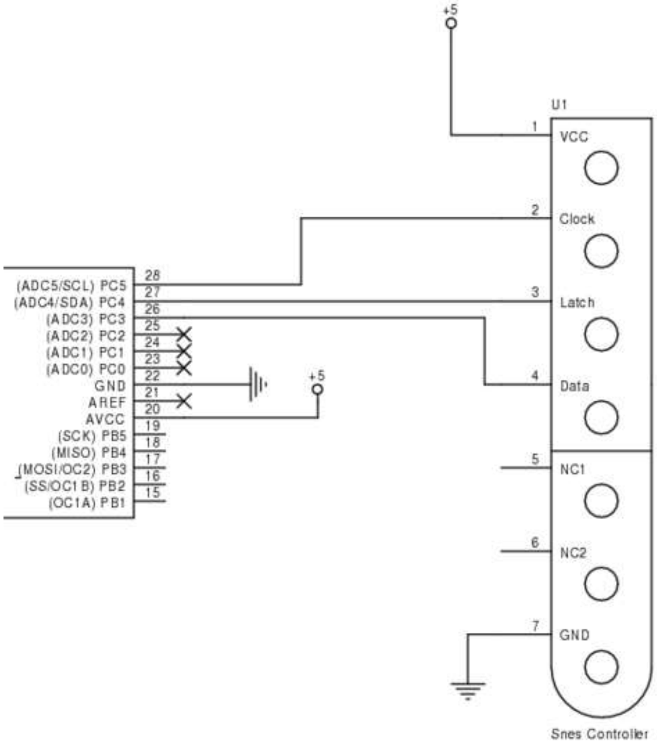


Image Notes
1. Your programmer will have these connections. Each programmer is different, so you will have to find a datasheet for you programmer to find the pins.
2. Atmega8 Chip Note the curve at the top. If you put the chip in backwards it will not work.
3. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
4. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
5. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
6. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
7. 12MHz Oscillator goes here. It can be connected either way.
8. Unused pins. I cut them off with a dremmel to save space.
9. More unused. Can be cut off.
10. Connect to programmer. Can be cut off after programming and testing.
11. Connect to Programmer. Can be cut off after programming and testing.
12. Unused Pin
13. Unused pins. Keep or cut off. Your choice.
14. Unused pins. Keep or cut off.
15. SNES Clock
16. SNES Latch
17. SNES Data
18. USB Data -. Also Zener Diodes and Resistors. See full schematic.
19. USB Data +. Also Zener Diodes and Resistors. See full schematic.

Step 5: Oscillator + SNES

The oscillator connects to pins 9 and 10. It can be rotated either way and work.
SNES Clock connects to pin 28.
SNES Latch connects to pin 27.
SNES Data connects to pin 26.
SNES 5V power connects to both 5V on the chip and usb 5V.
SNES ground connects to both grounds on chip and usb ground.



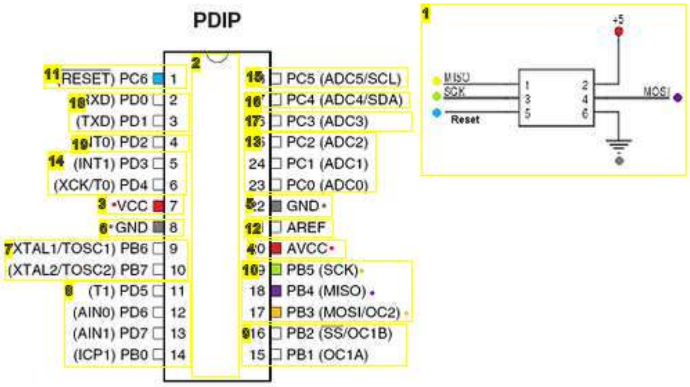
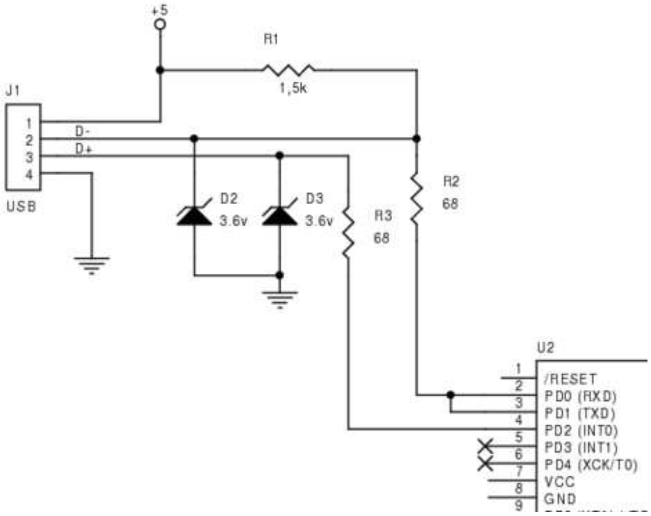


Image Notes
1. Your programmer will have these connections. Each programmer is different, so you will have to find a datasheet for you programmer to find the pins.
2. Atmega8 Chip Note the curve at the top. If you put the chip in backwards it will not work.
3. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
4. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
5. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
6. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
7. 12MHz Oscillator goes here. It can be connected either way.
8. Unused pins. I cut them off with a dremmel to save space.
9. More unused. Can be cut off.
10. Connect to programmer. Can be cut off after programming and testing.
11. Connect to Programmer. Can be cut off after programming and testing.
12. Unused Pin
13. Unused pins. Keep or cut off. Your choice.
14. Unused pins. Keep or cut off.
15. SNES Clock
16. SNES Latch
17. SNES Data
18. USB Data -. Also Zener Diodes and Resistors. See full schematic.
19. USB Data +. Also Zener Diodes and Resistors. See full schematic.

Image Notes
1. 12MHz Crystal

Step 6: USB Data and Power

This part is very tricky to wire correctly. Pay close attention to the schematic.
USB 5V goes to both 5V pins and SNES 5V
USB Ground goes to both ground pins and SNES ground
USB2 is data - . It goes:
through a 3.6v zener diode to ground
through a 68ohm resistor to pin 2 and 3
to usb 5V through a 1.5k resistor.
USB3 is data +. it goes:
through a 3.6v zener diode to ground
through a 68ohm resistor to pin 4



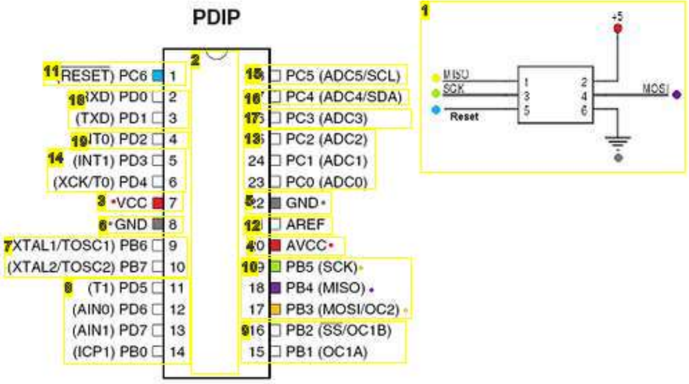
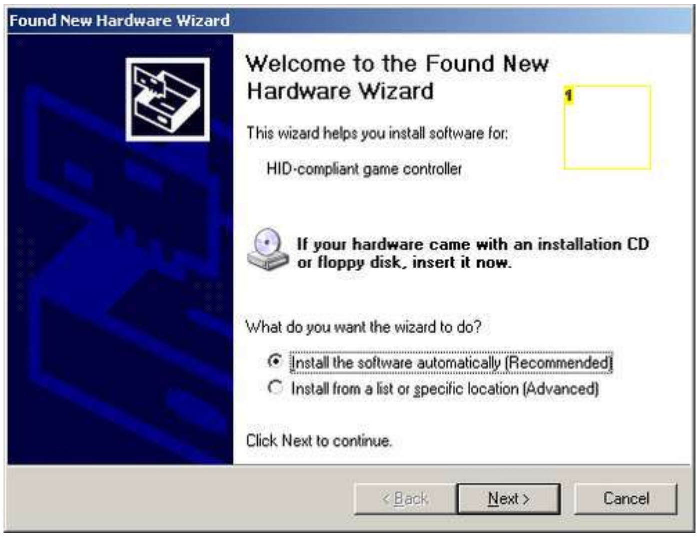


Image Notes
1. Your programmer will have these connections. Each programmer is different, so you will have to find a datasheet for you programmer to find the pins.
2. Atmega8 Chip Note the curve at the top. If you put the chip in backwards it will not work.
3. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
4. Connect all these: Pins 7, 20; USB 5V power; SNES 5V power; Programmer 5V power;
5. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
6. Connect all these: Pins 8, 22; USB Ground; SNES Ground; Programmer Ground;
7. 12MHz Oscillator goes here. It can be connected either way.
8. Unused pins. I cut them off with a dremmel to save space.
9. More unused. Can be cut off.
10. Connect to programmer. Can be cut off after programming and testing.
11. Connect to Programmer. Can be cut off after programming and testing.
12. Unused Pin
13. Unused pins. Keep or cut off. Your choice.
14. Unused pins. Keep or cut off.
15. SNES Clock
16. SNES Latch
17. SNES Data
18. USB Data -. Also Zener Diodes and Resistors. See full schematic.
19. USB Data +. Also Zener Diodes and Resistors. See full schematic.

Step 7: Testing the Controller

With all parts in place, plug in the usb port into your computer and it should come up with found new hardware. You can test all the buttons in the control panel under game controllers.

If it works, you are ready to transfer it to a pcb and put it in the controller.


Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

HID-compliant game controller

If your hardware came with an installation CD or floppy disk, insert it now.

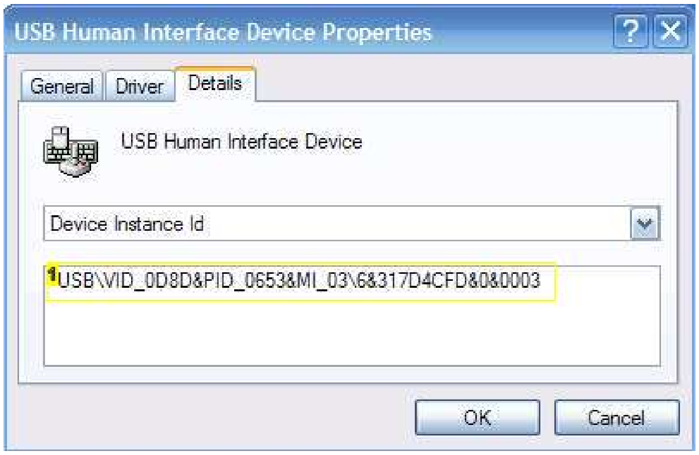
What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.

Image Notes
1. If you see this message, tell it automatic. I have not seen it on any of the machines I tested.


USB Human Interface Device Properties

General | Driver | Details

USB Human Interface Device

Device Instance Id

USB\VID_0D8D&PID_0653&MI_03\6&317D4CFD&0&0003

Image Notes
1. Google image. Not this device.

Step 8: Breadboard to PCB

I used the board below to make my design. I put the chip in place and used a dremmel to cut all unneeded areas off the board. You need to make it as small as possible to fit in the controller.

Note that all the solder spots near the pins connect 3 holes. This allows 2 wires per pin to be used on this board.

Note. The solder must be on the bottom of the board. Put it on the wrong side and the pins won't make any connections.

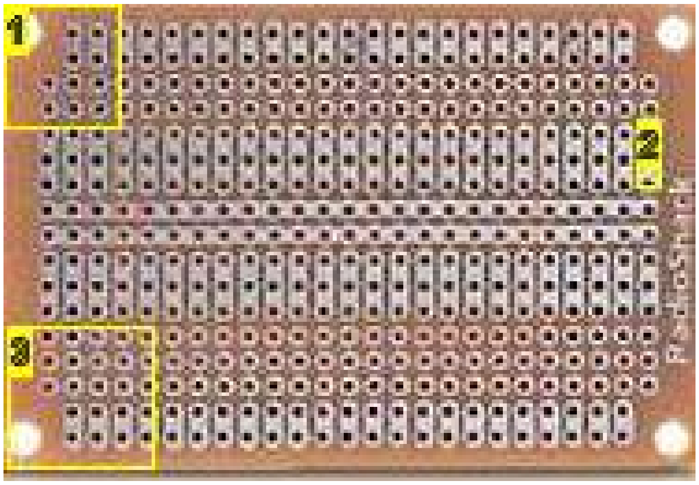


Image Notes
1. PCB. This board can be found at Radioshack for $2. It needs to be cut down to fit in the controller.
2. This pin can have 2 wires connected to it.
3. The atmega8 chip should not be on this side.

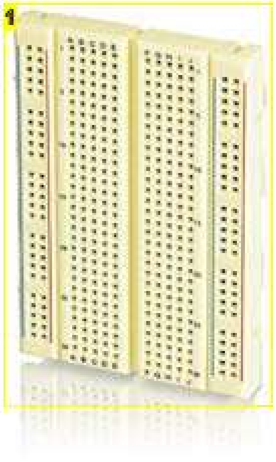


Image Notes
1. A small board is all you need. A bigger one will work, but it costs more and takes up alot more desk space.

Step 9: Modifying the Controller

I had to cut the casing a little to fit everything inside. I put my circuit under the abxy buttons. The picture shows what I did. Fit it in anyway you want as long as it closes.

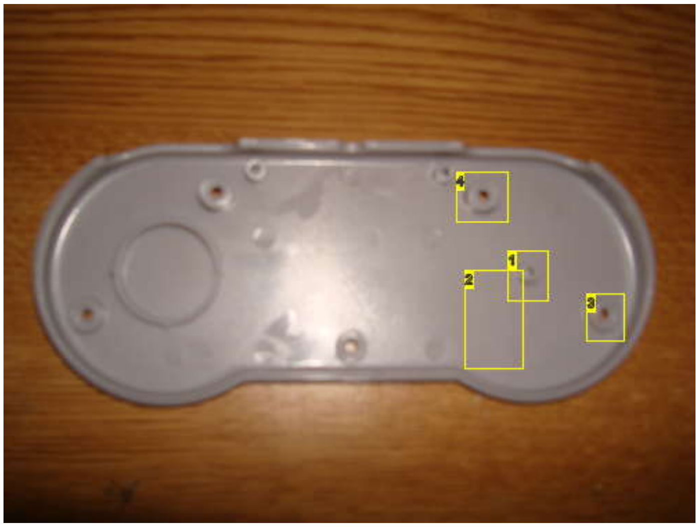


Image Notes
1. cut off
2. Circuit fits here diagonally.
3. Do not remove. Case screw.
4. Do not remove. Case screw.

Image Notes
1. Unmodified SNES controller inside.

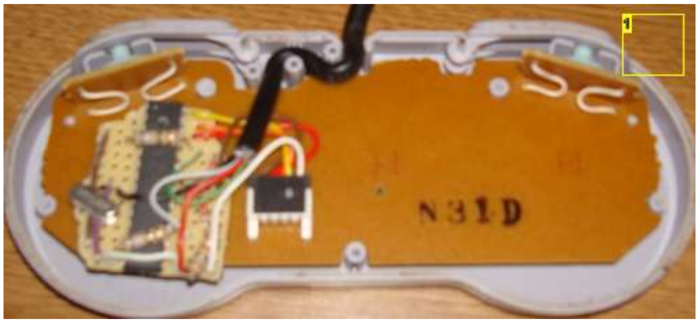


Image Notes
1. The way I setup my circuit.

Step 10: Final Testing + Notes

Test it and see if it works. If it does, you're done. If not check common errors.

Common Errors:

Check that power and ground are connected properly
Check the direction on the zener diodes. They do not work both ways.
Is the oscillator connected?
Did you program the right program and the fuse bytes?
Are snes data, latch, and clock hooked up to the right pins?
Do you have a solder bridge anywhere?

NOTES:

This controller does work on a PS3, but the buttons do not map very nicely. The start and select buttons do not map to ps3 start and select. This can be usable on some fighter games that allow you to remap the buttons.

Let me know what I can do to improve this Instructable.

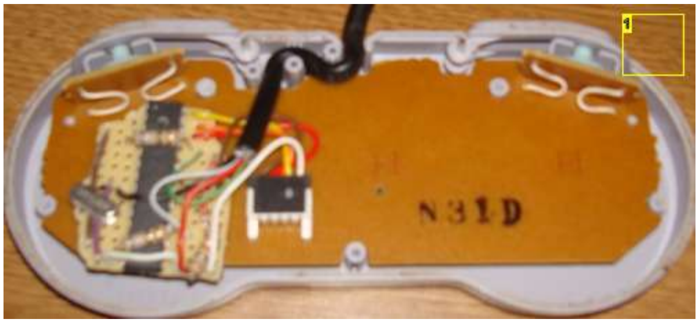


Image Notes

1. The way I setup my circuit.

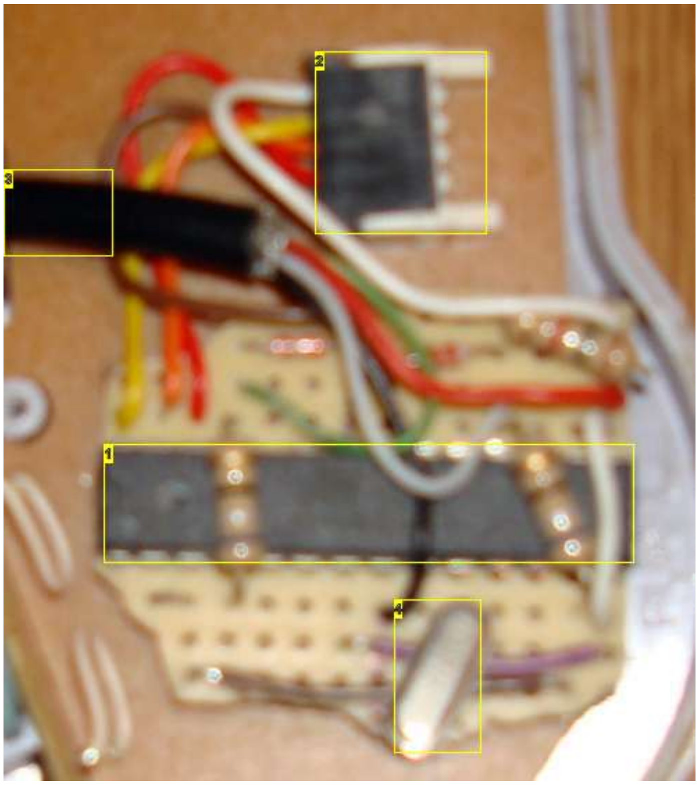


Image Notes
1. Atmega 8
2. Snes Wire Socket
3. USB Cable
4. Oscilator

Related Instructables

SNES to Parallel Port by Hungry_Myst

How-To : Make a NES to SNES adapter by thisissafety

SNES Controller into a Mouse Mod by theurbanraptor

SNES USB controller and flash drive by Nonentity

Snes cartridge usb hub by kristopher.hammar

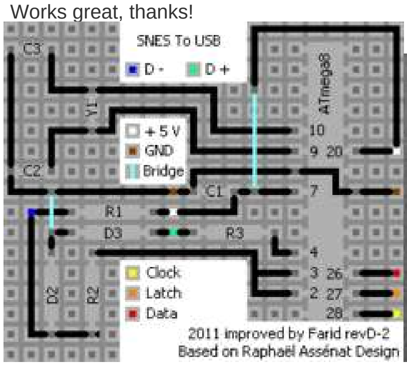
How To Install an SNES cable onto a Multi-Console Cthulhu by Toodles

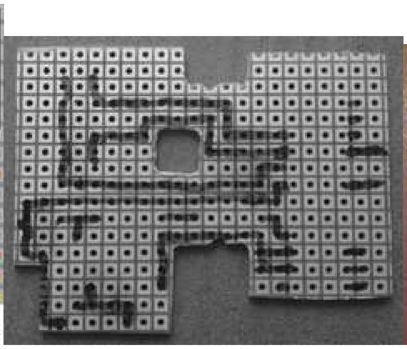
Comments

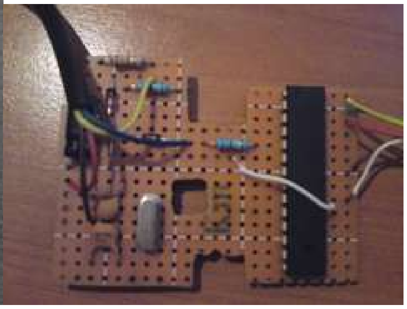
19 comments **Add Comment**

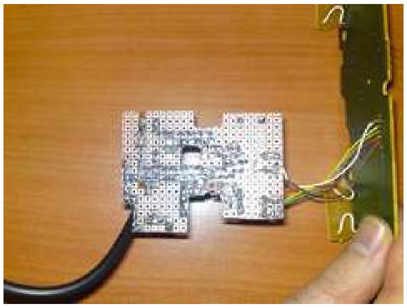
krauser35 says: Mar 10, 2011. 3:39 AM **REPLY**

Works great, thanks!









krauser35 says: Mar 6, 2011. 11:48 PM **REPLY**
What about C1? Did you forget it?
C1: 10uf capacitor. Install it near the ATmega8

http://www.upload-jpg.com/images.php/2e254839/01.PNG

Or it is just an unnecessary component, just like J1, J2, D1, R4?. But it doesn't seem to be unnecessary like these ones. Please explain about it. Thanks

ertxz18 says: Sep 6, 2010. 7:38 PM **REPLY**
Could you please include maybe some links or information as to how you acquired all the parts needed :D THANKS!

timeblade0 says: Oct 20, 2010. 1:03 PM **REPLY**
Atmega 8 can be found on ebay by searching "Atmega8 pu"
AVR programmer can be found on ebay. search "usbasp"
12Mhz crystal - sparkfun.com
3.6V zener diode. ebay. search "3.6V zener diode"
Resistors from sparkfun, radioshack,or ebay.
USB male from sparkfun, radioshack, ebay, or thrift stores. Any device with a usb male can work.
SNES controller - sparkfun, ebay, thrift stores, old game stores.
Wire - sparkfun, radioshack. Look for a jumper wire kit such as:
http://www.sparkfun.com/commerce/product_info.php?products_id=124
PCB and breadboards can be found at sparkfun, radioshack,or ebay.

Let me know if you have trouble locating the parts.

LordGormyr says: Jun 14, 2010. 11:30 PM **REPLY**
was wondering can this be done with a gamecube controller

timeblade0 says: Jun 15, 2010. 7:07 PM **REPLY**
Yes it can. This can be done with: NES, SNES, N64, GCN, Genesis, and Atari. I have done NES, SNES, and N64 successfully.
http://www.raphnet.net/electronique/electronique_en.php ^ has the circuit design. It does not cover putting it into the controller. I might make a guide on that sometime.

Pie Ninja says: Jun 28, 2010. 11:02 PM **REPLY**
Do you know whether it's possible to have more than 2 ports on the GCN/N64 to USB adapter? Like, wire 4 controller ports to the one board?

timeblade0 says: Jul 3, 2010. 12:33 PM **REPLY**
Possible yes. Easy no. I don't know exactly how. 2 ideas: 1: make a 4 port usb hub and attach 4 n64/gc to usb circuits to it. I have seen guides to make usb hubs with micro controllers on the web. 2: the program is open source. recode it to have 4 controllers in and 1 usb out. This method is probly very hard. The site has a guide for how to do it with an SNES/NES 4 port. Might be similar for N64/GC.

technoguy94 says: Apr 28, 2010. 5:46 AM **REPLY**
So its used as an actual USB game controller, instead of just another emulated keyboard?

timeblade0 says: Apr 28, 2010. 8:23 AM **REPLY**
Yes, this uses a standard HID controller driver. Windows, mac, Linux, and PS3 all have built in support for this driver.

I did not write the code to handle the driver. This instructable is my attempt in using someone else's circuit inside an actual controller.

I do not like the keyboard emulation controllers.You can only press 2 or 3 buttons at a time, and sometimes it does not respond correctly.

XGundam05 says: Mar 23, 2010. 3:41 PM **REPLY**

This circuit will work in Linux. I modded a PC a while back to be a dedicated SNES emulator with 4 controller ports. Runs 8.04? Ubuntu using Flux Box.

andreq says: Jan 6, 2010. 11:08 AM **REPLY**

I did the same thing some times ago, but I used an ATtiny45 (DIP).

The schematic and source code can be found here :
hobbyelektronik.org/w/index.php

I did some modification to the source code to include automatic internal OSC calibration. I'll try to find it somewhere on my hard drive.

If my memory is good, it might even fit inside an ATtine25 (that's what I had around...)

Anyway, great job, but you should show us how the finished PCB looked like and use less "marketing" pictures :)

timeblade0 says: Jan 6, 2010. 1:04 PM **REPLY**

I posted 2 pictures of my actual circuit.

Using an ATtiny25 would be a lot easier than the Atmega8 I used. Atmega8 = 28 pins, ATtiny25 = 8 pins, much easier to cram in the case. I will try the one you posted on my next attempt at SNES USB.

andreq says: Jan 6, 2010. 4:07 PM **REPLY**

This is the main reason why I used the ATtiny... you also have less component!

The only problem I had was with the USB data voltage.

I tryed the zener diode method without success.

On the V-USB website (the actual USB "firmware" that is used by this code) you can see other method.

For my circuit I used a 3mm red led on the 5v line... it reduced the voltage close to the 3.6v required by the data line. I also removed the zener.

I think 2 "normal" (not led...) diode on the 5v is also possible.

andreq says: Jan 6, 2010. 12:11 PM **REPLY**

Here is the full sources.

Include schematic, code and compiled HEX.

www.filemonster.net/en/file/15003/Snes-Attiny-zip.html

It should fit inside an ATtiny25, it compile to 1984 bytes (kinda... funny)

blackowaya says: Mar 13, 2011. 8:14 PM **REPLY**

Hi!

I'm afraid the link to the firmware made by andreq is dead.

I'm trying to make this adaptor, using an ATTINY25, i tried to add internal OSCCAL and optimize the code from hobbyelektronik but my firmware compile to 2138 bytes (it's too fat).

If someone have the source code of the firmware of andreq, please post it. Thank you in advance.

andreq says: Mar 13, 2011. 9:46 PM **REPLY**

Lucky enought, I found it inside an old backup folder :

http://www.2shared.com/file/1I9Vq85d/Snes-Attiny.html

You might have to look at the code. I've seen I commented out the autocalibration part (to save space). Also, I changed the usb vendor name to Andre in the usb-config. You could just change that to anything small (<=5 letters) if you care. Make sure to set the lenght accordingly.

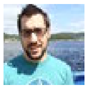
You might need to play with the OSCALL value which is read from eeprom 0x00. It should be between 125 and 128.

Have fun! There's nothing like the real feeling of a Snes game-pad when playing classic games.

blackowaya says: Mar 14, 2011. 6:36 PM **REPLY**

WOW, thanks a lot andreq ¡n_n¡.

I'll be testing it tonight. Yeah, nothing better to use the original SNES pads to play classic games. I'm a SNES fan and i think is the best pad in terms of weight, confort, response and button layout. I'll be posting photos when finish the project.



andreq says: Mar 13, 2011. 9:30 PM **REPLY**

I'll try to find that file on my computer again and upload it somwhere else. If your in a hurry, the easiest part to "cut" is the USB device name. It take lots of space.