10 Ways to Destroy An Arduino

Introduction

Use a sledgehammer, fire a bullet at it, throw it into a pool....that's not what we're talking about. We're going to show you how to electrically destroy your Arduino, though many of you seem to already know how to do that through unfortunate experience. You know what we mean....that funny smell, the scorch mark on a component, or the dreaded "programmer not in sync" error message -- all signs that you've just learned a lesson the hard way.

Why are we doing this? If you own an Arduino, it's good to know what is and what isn't OK to do with it. We also want you to consider buying our Ruggeduino, which will survive all of the tortures described below.

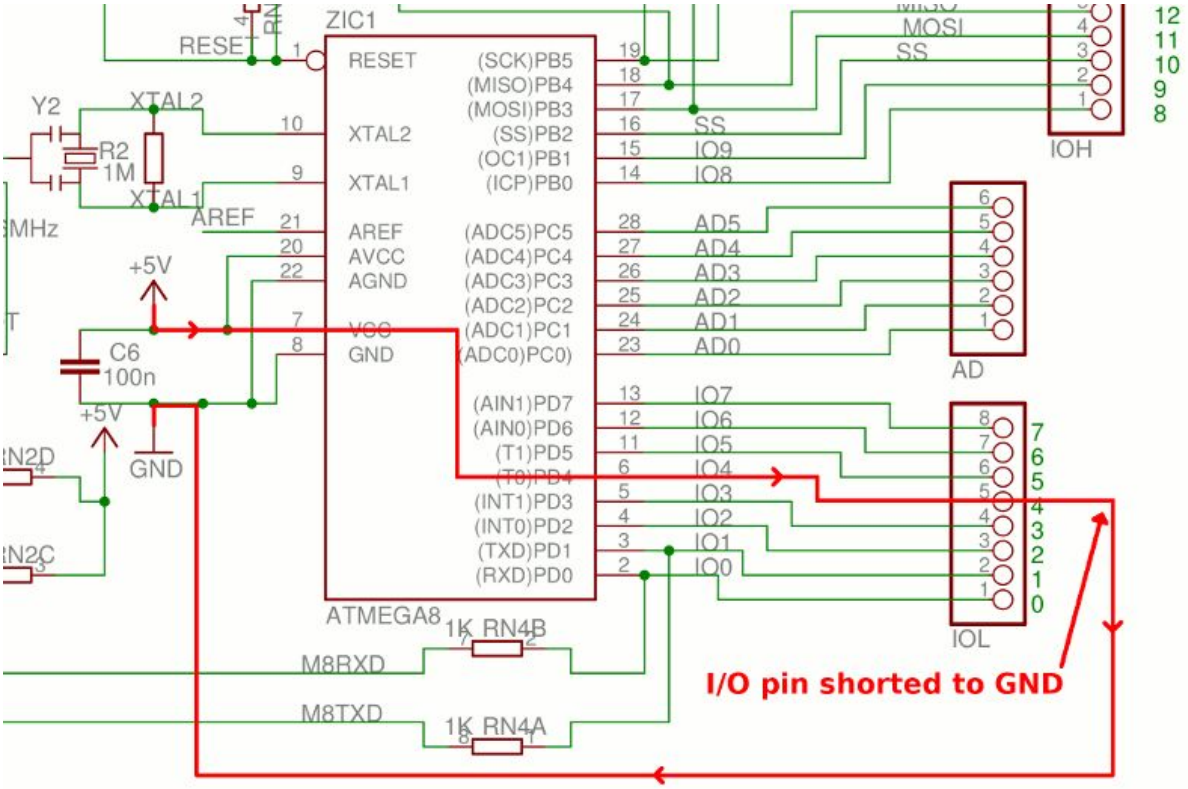
Method #1: Shorting I/O Pins to Ground

HOW

Configure an I/O pin to be an output then set it high. Short the pin to ground. You have now created an overcurrent condition on the I/O pin and it will be destroyed.

WHY

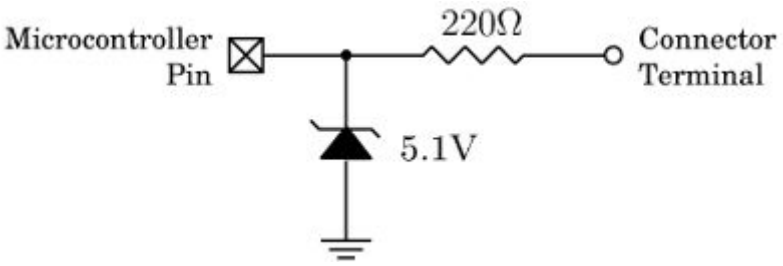
Here is the path of current flow (the schematic is for the Arduino Uno, which can be found here):



The microcontroller datasheet specifies an absolute maximum per-pin current of 40mA. With a typical internal resistance of only 25 ohms per pin, a dead short to ground can allow as much as 200mA of current to flow, more than enough to destroy the microcontroller pin.

THE FIX

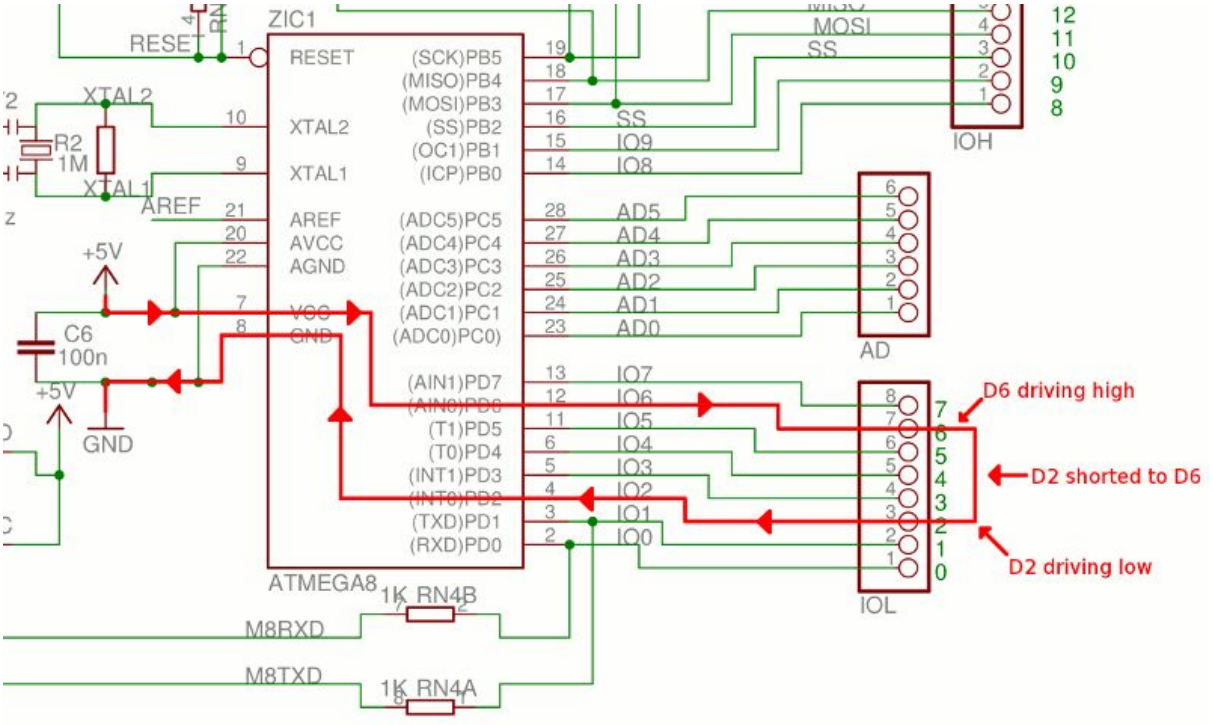
The Ruggeduino protects against this destruction by putting a 30mA resettable fuse (PTC) in series with every I/O pin. Not only is the current safely limited to 30mA under all conditions (more on this below), but the built-in 220 ohm resistance of the fuse naturally limits the current to 5V/220 = 23mA right off the bat.



Method #2: Shorting I/O Pins to Each Other

HOW

Configure two I/O pins to be outputs then set one high and the other one low. Now connect the pins together. You have now created an overcurrent condition on both I/O pins and they will be destroyed.



WHY

The path of current flow is similar to Method #1 above except the ground return path is through the microcontroller.

THE FIX

Same as for Method #1, the Ruggeduino protects against this destruction by putting a 30mA resettable fuse (PTC) in series with every I/O pin.

Method #3: Apply Overvoltage to I/O Pins

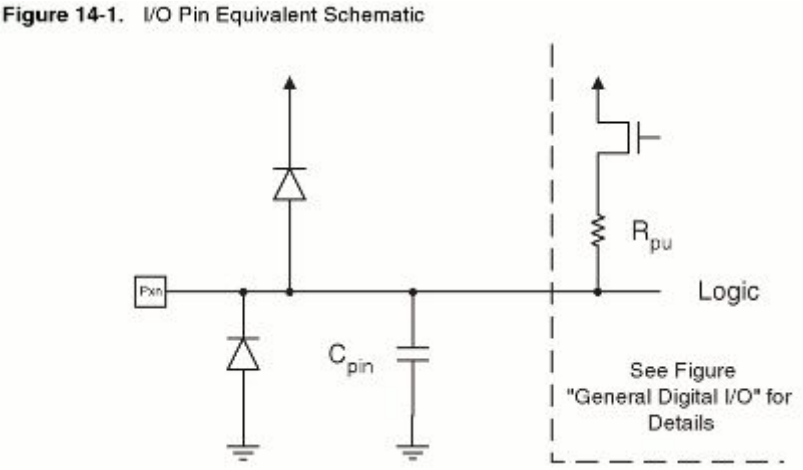
HOW

Apply a voltage exceeding 5.5V to any I/O pin. The I/O pin is destroyed.

WHY

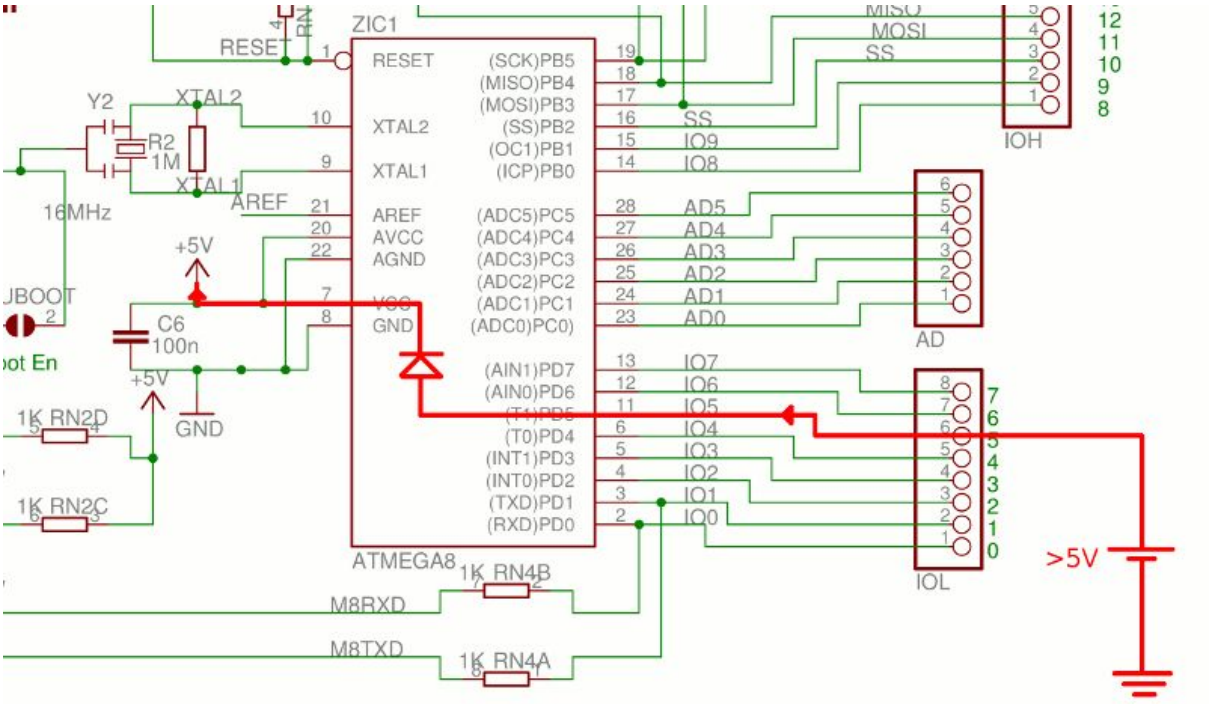
This method of destruction forward-biases the ESD protection diode built-in to the microcontroller. Here is a model of each microcontroller I/O pin from the Atmel ATmega328P datasheet:

Figure 14-1. I/O Pin Equivalent Schematic



Once the voltage at the I/O pin is greater than the supply voltage (5V) by about 0.5V, the top diode starts to conduct current. This is OK for diverting a short-duration overvoltage event, like ESD (electro-static discharge), but that diode is not meant to be on all the time. It will simply burn out and stop protecting the pin.

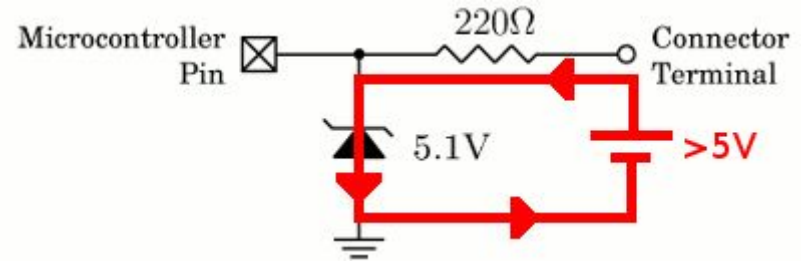
This diagram shows the flow of current when overvoltage is applied to an I/O pin.



If the internal protection diode fails open, then the overvoltage destroys the I/O pin. If the protection diode fails by shorting out, it's even worse because now the overvoltage is applied to the entire +5V supply on the Arduino. This means it will reach other components, like the USB interface chip, and destroy them too.

THE FIX

On the Ruggeduino, every I/O pin is protected by a 30mA resettable fuse (with built-in 220 ohm resistance) and a 5.1V zener diode that together serve to limit the pin voltage to 5.5V, regardless of applied overvoltage (up to 24V).



Now, instead of current flowing through the microcontroller's internal protection diode, it flows safely through the zener diode, to ground, and back to the source of the overvoltage. The PTC fuse limits this current to 30mA so the 5.1V zener diode does not dissipate excessive power.

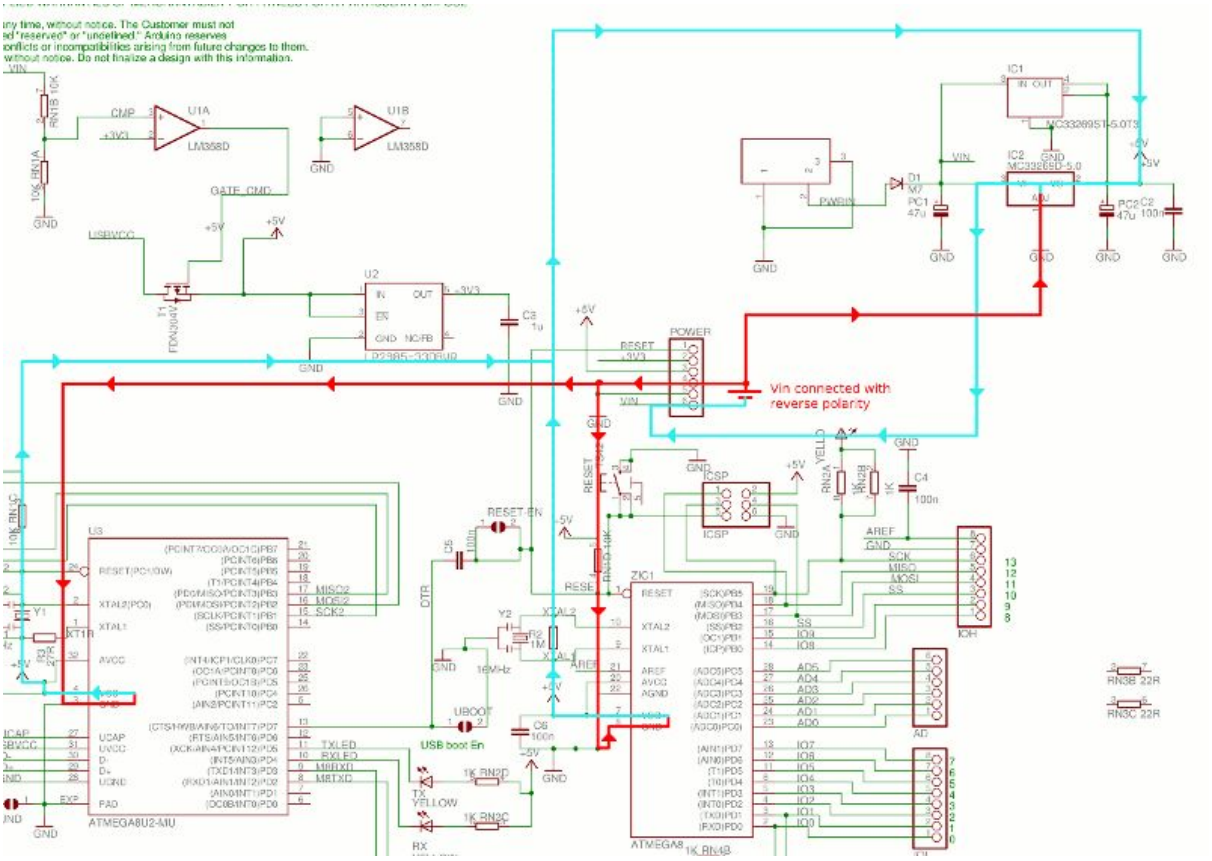
Method #4: Apply External Vin Power Backwards

HOW

Power your Arduino through the Vin connector pin, but reverse the polarity of the Vin/GND power connection. You will destroy several devices on the Arduino.

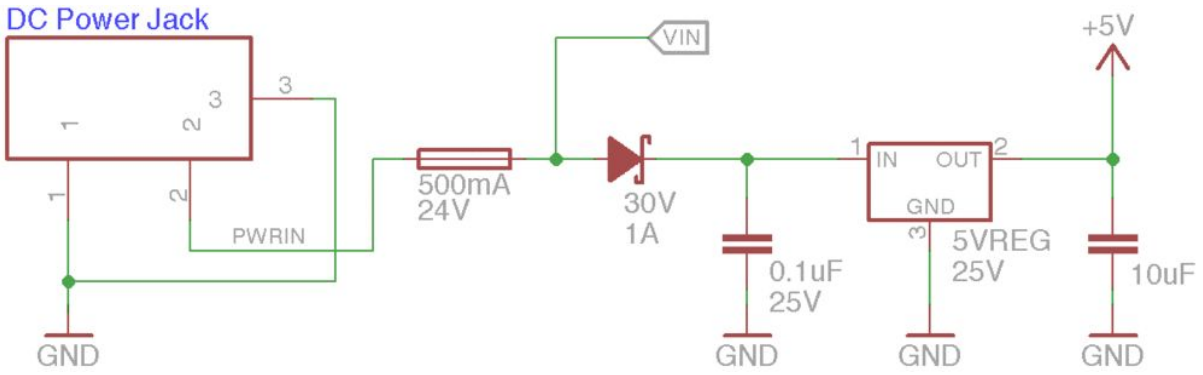
WHY

There is no reverse-voltage protection on voltages applied to the Vin connector pin. Current will flow from the GND pin of the ATmega328P back up through the 5V pin, back through the 5V regulator and to Vin. The same thing will happen with the ATmega16U2 microcontroller. Both microcontrollers and the 5V regulator will be destroyed.



THE FIX

On the [Ruggeduino] the Vin pin is protected by a 30V reverse-blocking diode, as shown in the schematic.



You can apply as much as 30V of reverse-polarity voltage on the Vin pin without causing any damage.

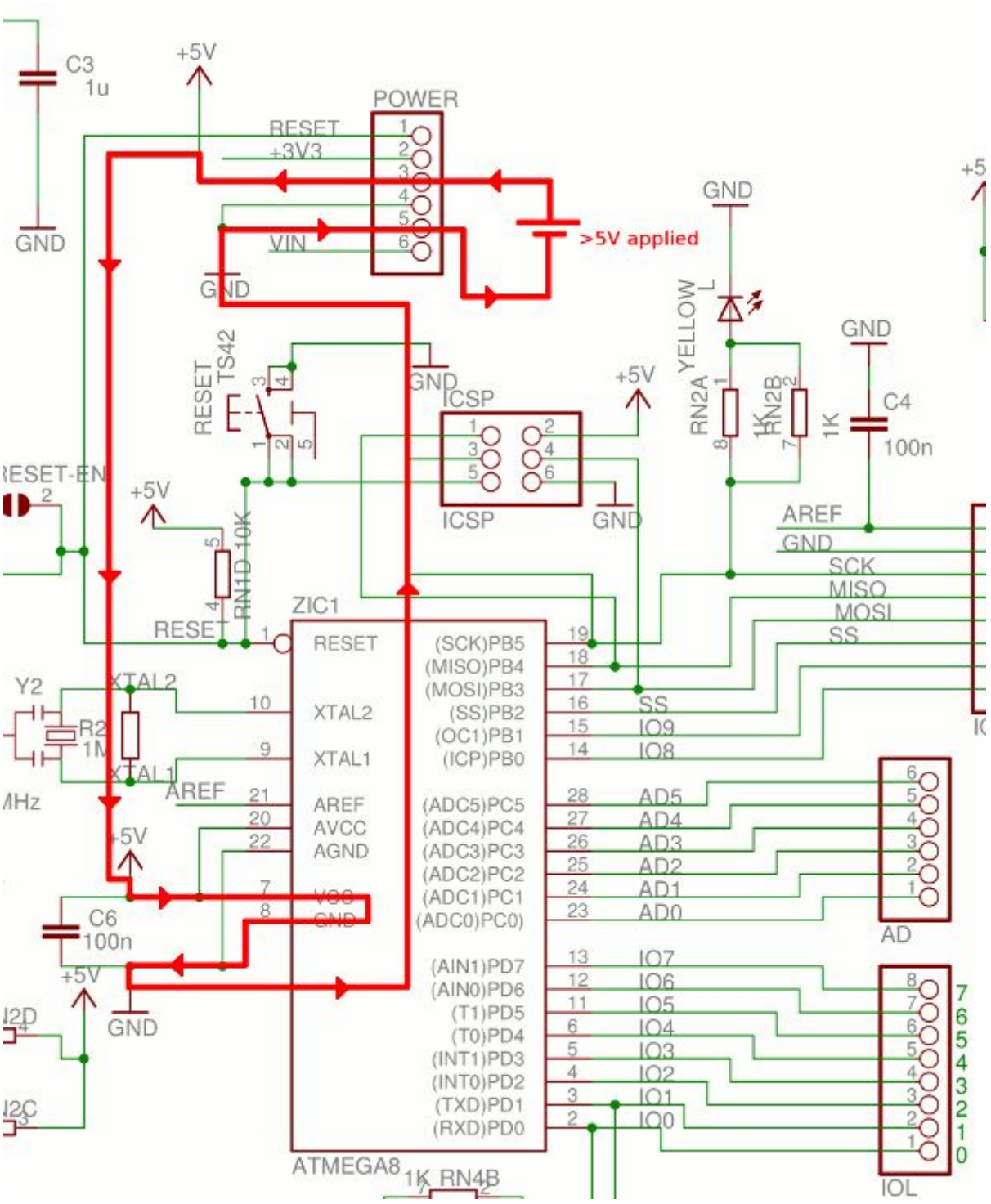
Method #5: Apply >5V to the 5V Connector Pin

HOW

Apply a voltage of 6V or higher to the 5V connector pin. Many components on the Arduino will be destroyed, and this voltage can also appear on your computer's USB port, possibly damaging it.

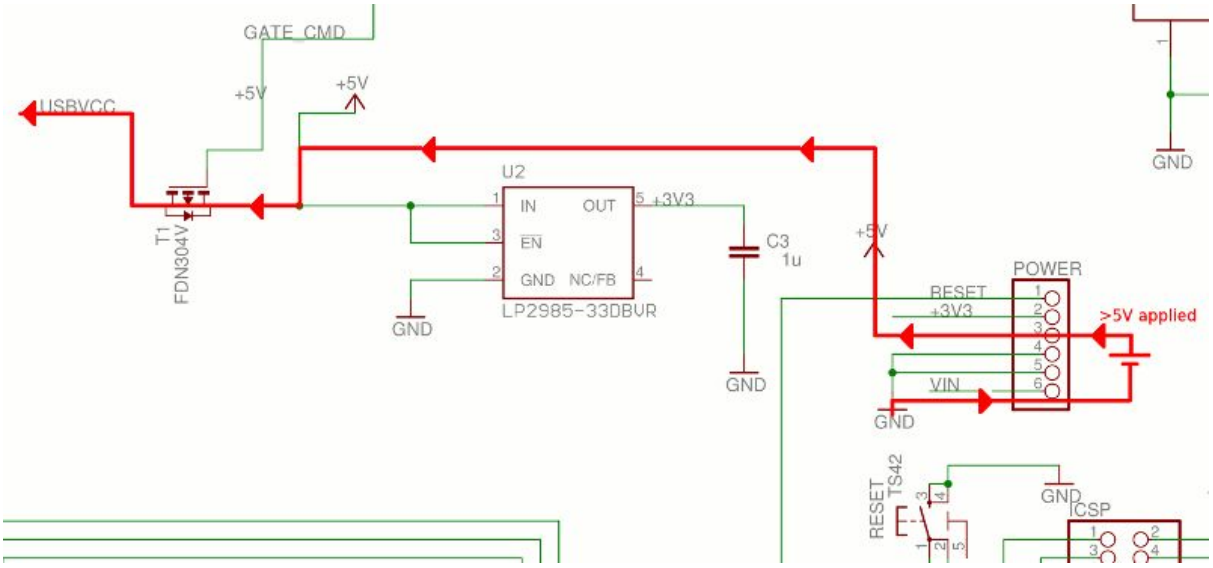
WHY

There is no protection on the 5V connector pin. This voltage is directly connected to the ATmega328P microcontroller, the ATmega16U2 USB interface microcontroller, and the 5V regulator, all of which can be damaged by voltages exceeding 6V, and the resulting currents that flow. Here is an example current path through the ATmega328P microcontroller.



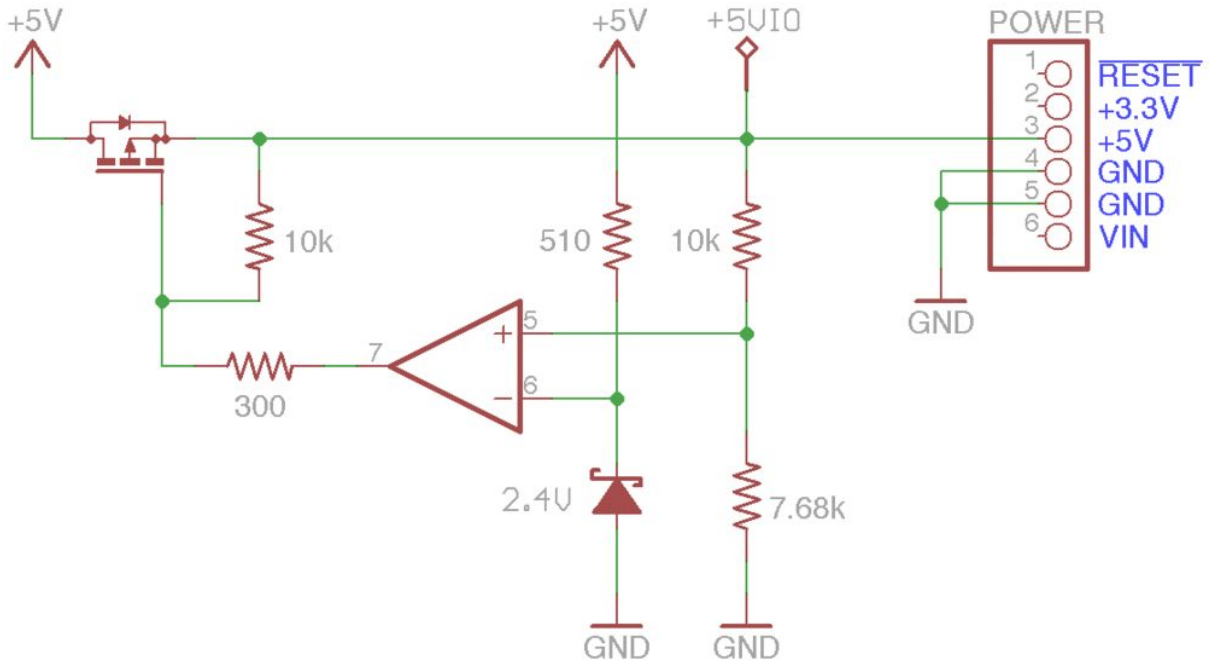
It is a common misconception that the Arduino 5V regulator will ensure that the 5V voltage remains at 5V, no matter what. IT WILL NOT! The only thing the 5V regulator can do is control current coming from the USB port or the external DC power jack. If the current is coming from an external power source directly connected to the 5V connector pin, the regulator can do nothing about it.

Another consequence of applying more than 5V to the 5V connector pin is possible damage to the PC's USB port. If the Arduino is powered from USB then this excessive voltage can cause current to flow backwards through the voltage-switching MOSFET T1 and back to the PC's USB port.



THE FIX

On the Ruggeduino a voltage cutoff circuit makes sure that the 5V connector pin is disconnected if it exceeds 5.5V.



You can apply up to 24V on the 5V connector pin and the Ruggeduino components will never see it, and nothing will be damaged.

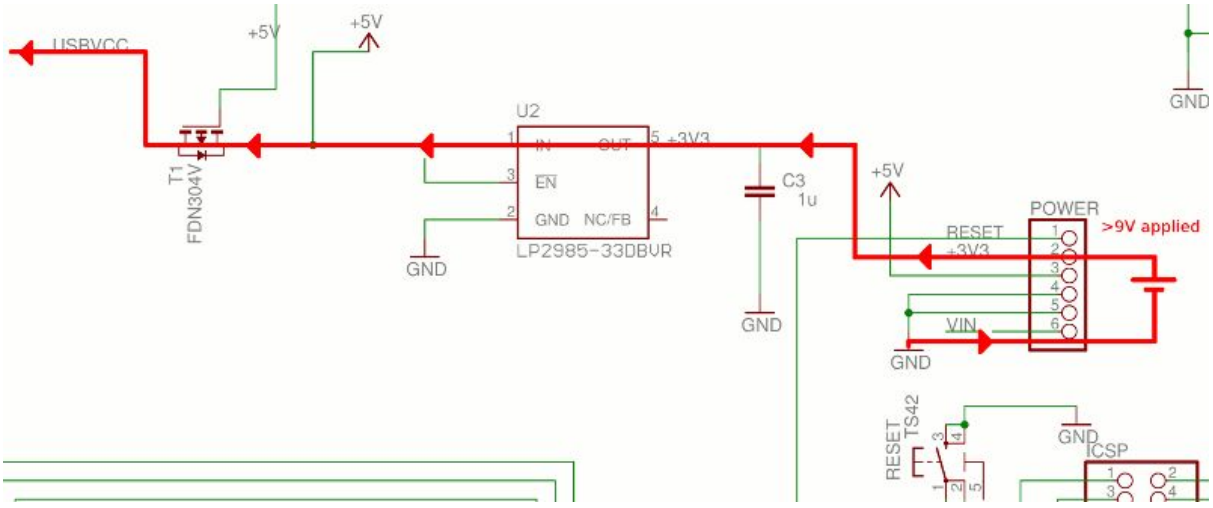
Method #6: Apply >3.3V to the 3.3V Connector Pin

HOW

Apply a voltage of 3.6V or higher to the 3.3V connector pin. Any 3.3V shields plugged in, or other devices powered from this pin, will be destroyed. If at least 9V is applied, this voltage can destroy the Arduino 3.3V regulator and also feed current back into the PC's USB port.

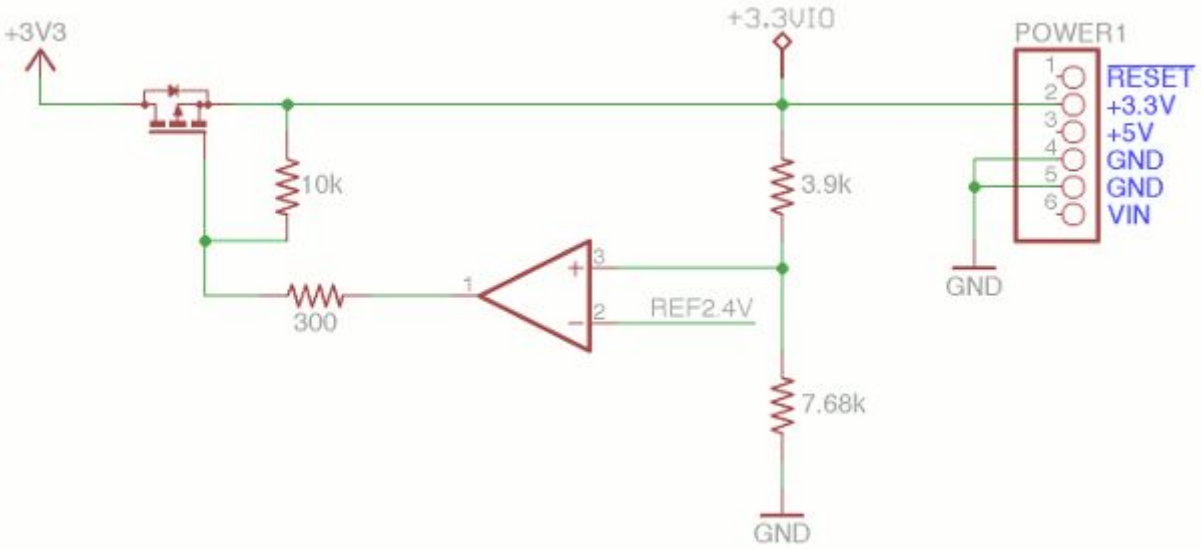
WHY

The 3.3V connector pin has no protection circuitry. This voltage is directly connected to the Arduino 3.3V regulator and any other shields or devices that are powered by this connector pin. If the voltage exceeds 9V, the 3.3V regulator will be destroyed and may allow current to flow backwards to the 5V node, and then backwards further to the PC's USB port. The excessive voltage will also destroy the two devices connected to the 5V node: the ATmega328P and ATmega16U2 microcontrollers.



THE FIX

Similar to the 5V protection circuit, the Ruggeduino has a voltage cutoff circuit for the 3.3V connector pin. Any voltage applied to this pin greater than 3.6V disconnects the pin from the Ruggeduino 3.3V node.



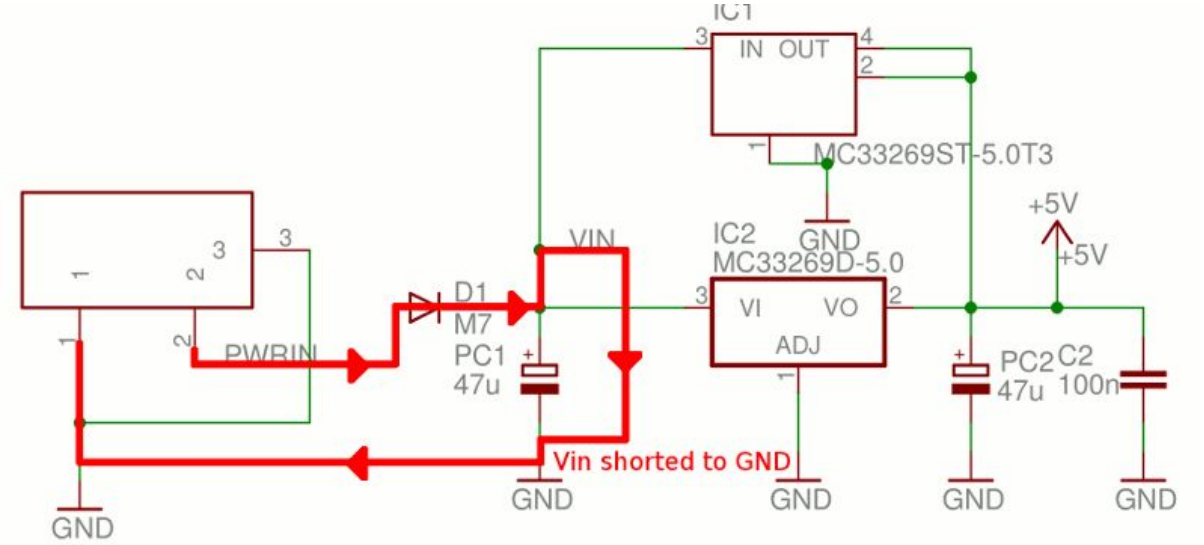
Method #7: Short Vin to GND

HOW

Power the Arduino from the DC power jack and short the Vin connector pin to GND. The Arduino blocking diode will be destroyed and traces on the Arduino PCB may melt and be destroyed.

WHY

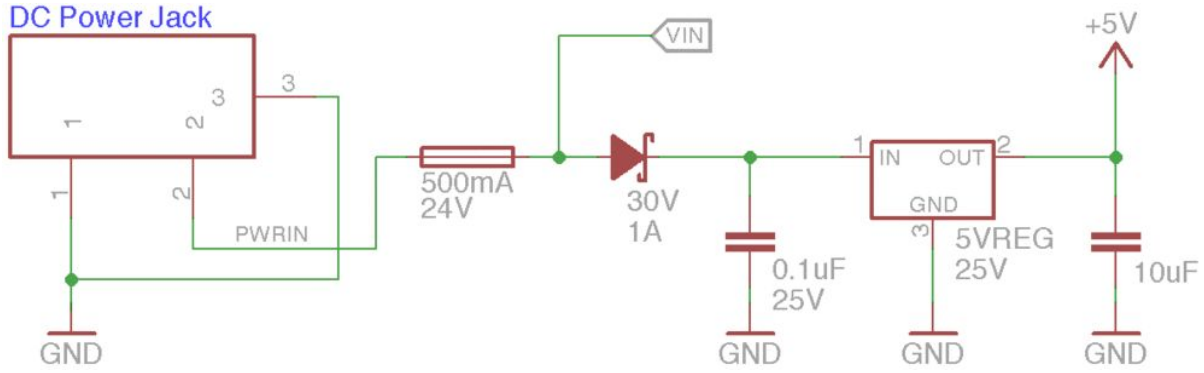
There is no current limit protection on the Vin connector pin. A short circuit from Vin to GND effectively short circuits the DC power jack input, and exceeds the current rating of the blocking diode.



The amount of current that flows is limited only by the resistance of the Arduino PCB traces and the current capability of the power supply. If this is high enough, the diode D1 will be destroyed and PCB traces may melt due to the heat caused by this large current.

THE FIX

The Ruggeduino has a 500mA PTC resettable fuse in series with the DC power input circuit (just like the one that protects the USB power input). This fuse limits the current to safe levels even if Vin is shorted to GND.



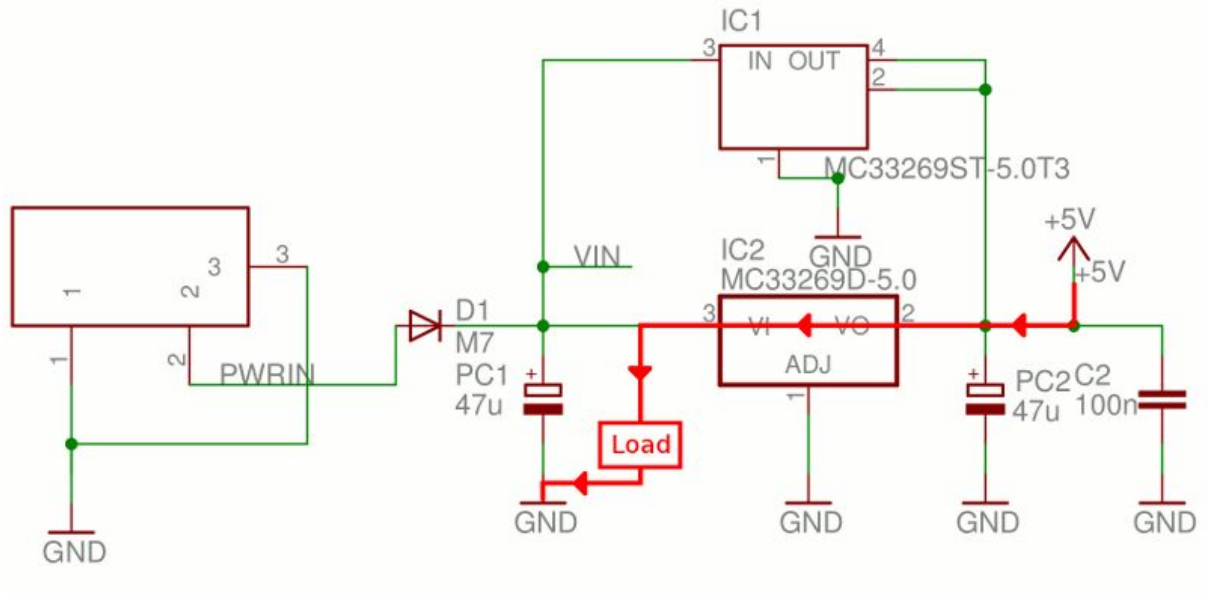
Method #8: Apply 5V External Power with Vin Load

HOW

If you are powering the board from 5V applied to the 5V connector pin and you have circuitry connected to the Vin pin (or have shorted Vin to GND) then current will flow backwards through the 5V regulator and destroy it.

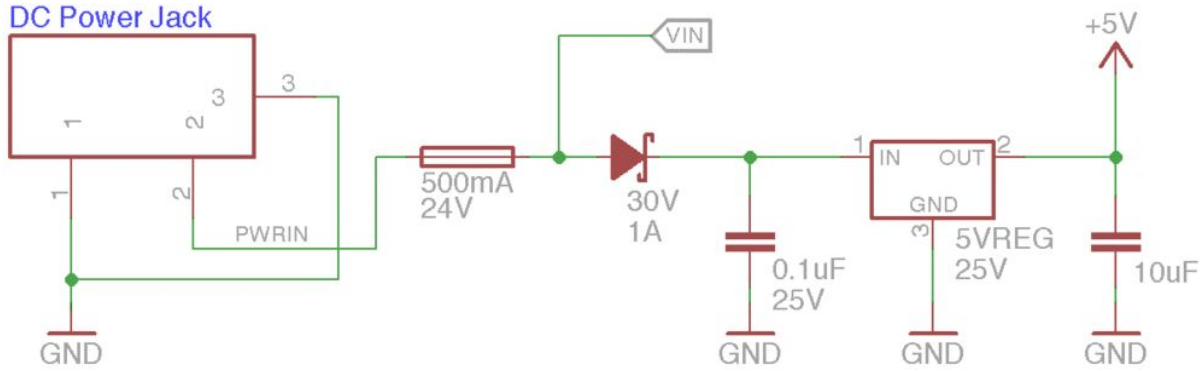
WHY

There is no reverse voltage protection on the 5V regulator thus current can flow from the 5V connector pin, backwards through the regulator, and to whatever is connected to Vin.



THE FIX

The Ruggeduino has its reverse-blocking diode right at the 5V regulator input, ensuring that no current can flow backwards through the regulator, even if a circuit is connected to the Vin pin.



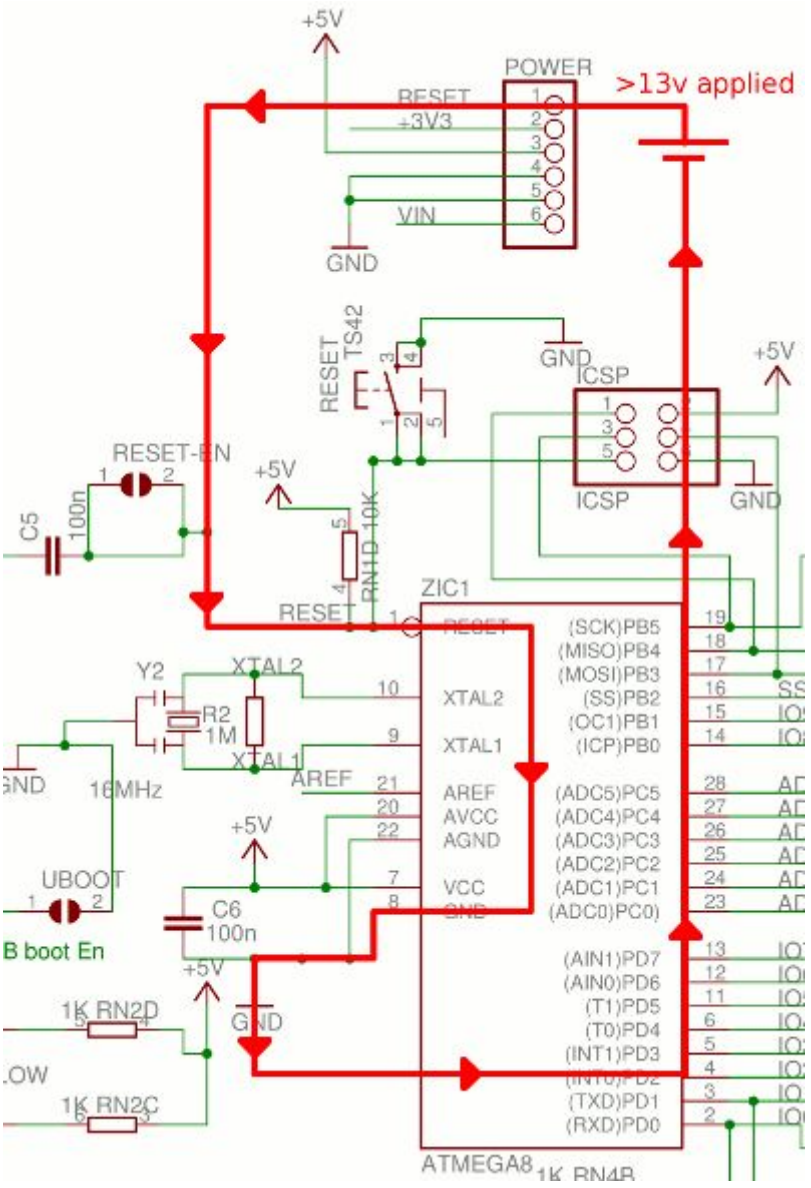
Method #9: Apply >13V to the Reset Pin

HOW

Apply >13V to the Reset connector pin. The ATmega328P microcontroller will be damaged.

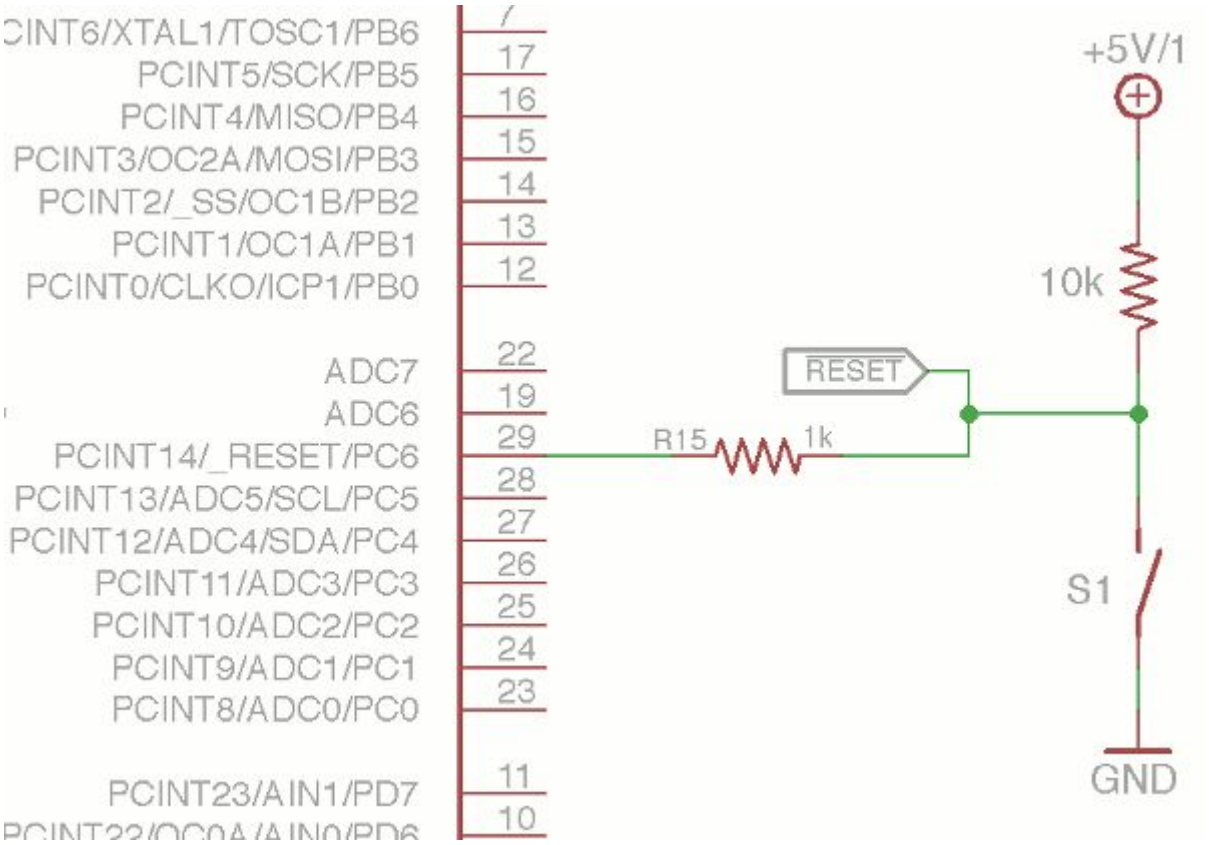
WHY

The Reset connector pin is directly connected to the reset pin on the ATmega328P. While this pin tolerates 13V, higher voltages will damage the device.



THE FIX

On the [Ruggeduino] a 1k resistor is placed in series with the ATmega328P reset pin. If voltages greater than 13V are applied to the Reset connector pin, this pin limits the current that can flow thus limits the damage to this pin.



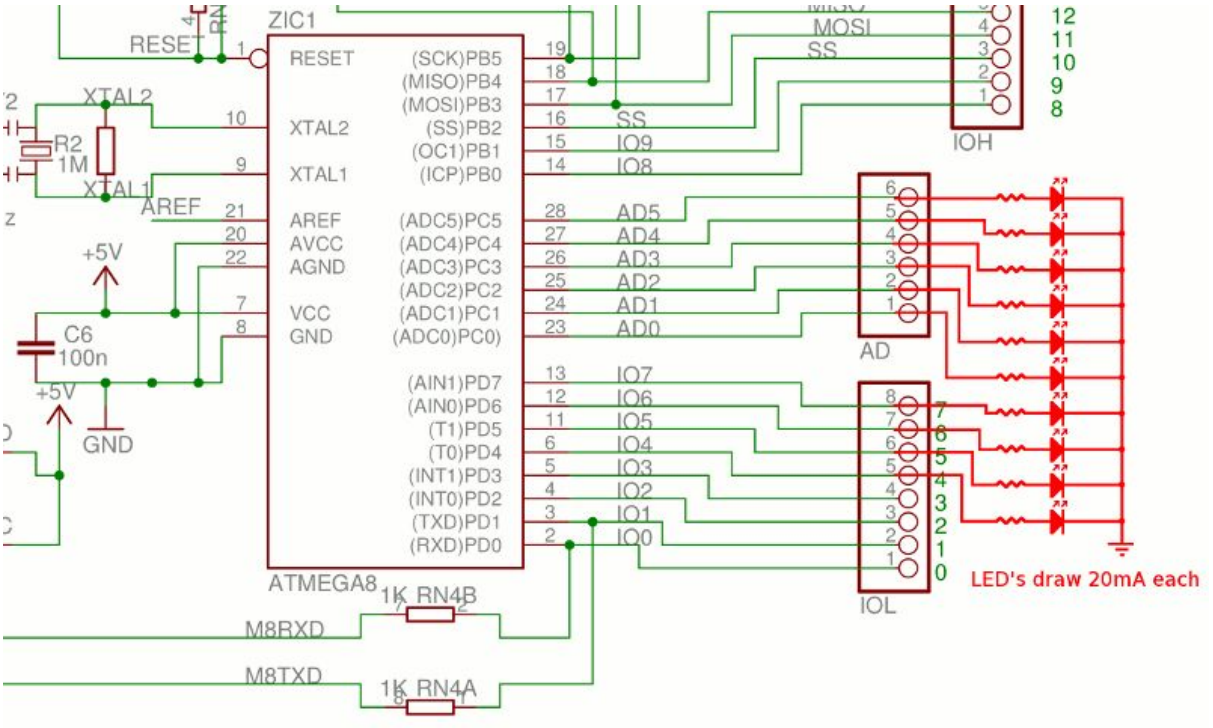
Method #10: Exceed Total Microcontroller Current

HOW

Configure at least 10 I/O pins to be high and draw 20mA from each one (for example, by lighting 10 LED's). You have now exceeded the total supply current rating for the microcontroller and it will be damaged.

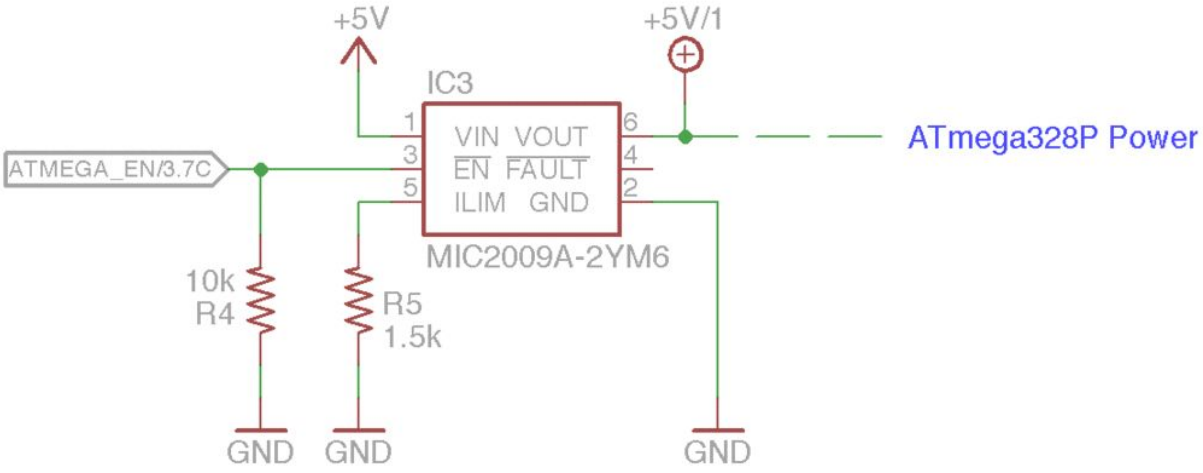
WHY

It's not enough to limit the current of each I/O pin -- the total current sourced from all I/O pins must not exceed 200mA, according to the ATmega328P datasheet.



THE FIX

On the Ruggeduino a dedicated current-limiting IC (MIC2009A shown below) ensures that no matter what currents you are sourcing from I/O pins, the total microcontroller supply current does not exceed 150mA (typical).



If more than 150mA of current flows to the ATmega328P, the MIC2009A starts reducing the voltage until the current is reduced to a safe level.