Intervalometer

by randofo

I decided to make a quality DIY intervalometer for my DSLR Pentax camera so that I could do time-lapse photography. This intervalometer should work with most major brands of DSLR cameras such as Nikons and Canons. It works by triggering the shutter using the camera's remote trigger port. It can also auto-focus before each shot if so desired (or toggle this on or off at any time). The brains of this intervalometer is an Arduino chip. It may seem very complicated at first glance, but is actually a simple circuit and not that hard to make.

Step 1: Go Get Stuff

You will need:

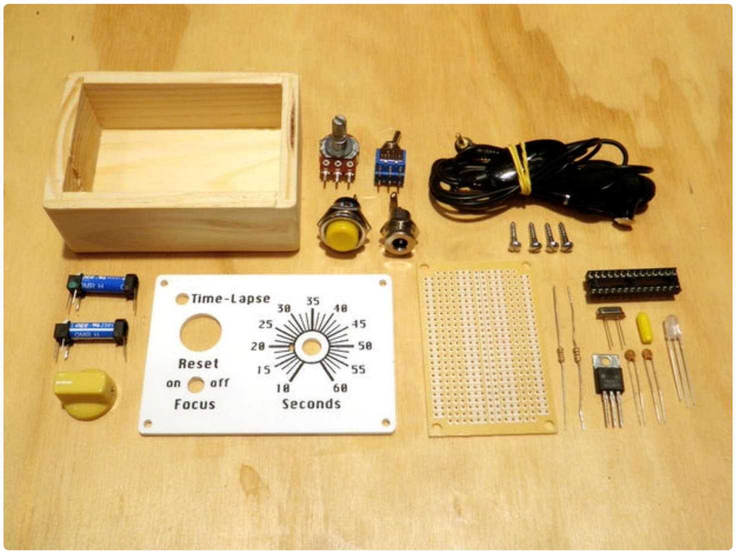
(x1) Small wood box
(x1) 1/8" acrylic panel (see next step)
(x1) black acrylic paint
(x1) paintbrush
(x1) Arduino Uno
(x1) PCB
(x1) 1K resistor

(x1) DC power socket
(x2) SPST 5V reed relays
(x1) shielded stereo cable
(x1) 3/32" (2.5mm) male plug
(x1) 9VDC power adapter
(x1) knob
(x1) toggle switch cover (optional)
(x4) 1" wood screws
(x1) red, black and green wire
(x1) soldering setup
(x1) multimeter

(x1) 100 ohm resistor
(x1) LM7805 5V regulator
(x1) 2-color LED
(x1) 16MHz crystal
(x2) 22pf capacitors
(x1) 10K potentiometer
(x1) 28 pin socket
(x1) DPDT toggle switch
(x1) SPST push-button switch

(x1) drill press (or hand drill) and misc. tools.

Some of the links on this page contain Amazon affiliate links. This does not change the price of any of the items for sale. However, I earn a small commission if you click on any of those links and buy anything. I reinvest this money into materials and tools for future projects. If you would like an alternate suggestion for a supplier of any of the parts, please let me know.



Step 2: Make the Front Panel

If you happen to have a 70 Watt Epilog laser cutter, do the following...

Download the attached template file. Put your 1/8" white acrylic in the machine (do not remove the protective coating). Adjust the border of the template appropriately to match the borders of your box.

Laser etch the design with the following settings:
power: 70
speed: 100
passes: 2

Cut out the template as such:
power: 100
speed: 9
frequency: 5000

When you are done lay down 2 - 3 thin coats of black paint and wait for them to dry before you peel off the protective coating from the acrylic. Use a craft knife to carefully pick off the bits of covering that remain.

Okay, okay... I know most of you don't have a laser cutter. Here are some alternatives:

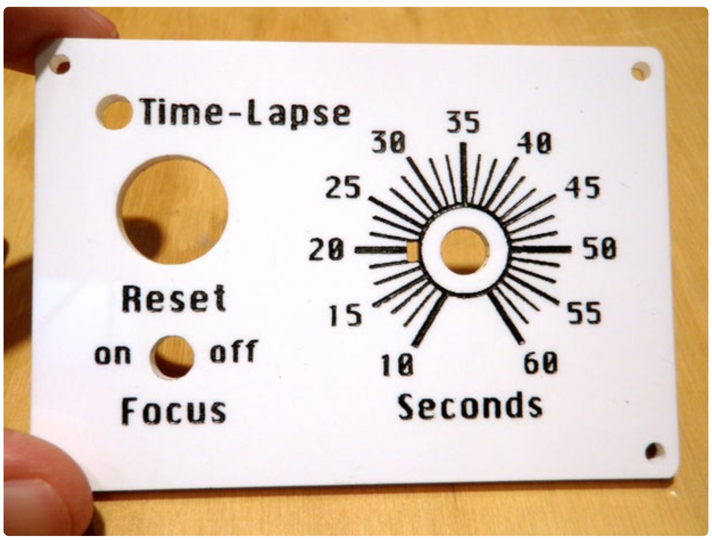
1) Download the file print out the design as a decal and also use the design as a template to cut out the panel with more traditional tools.

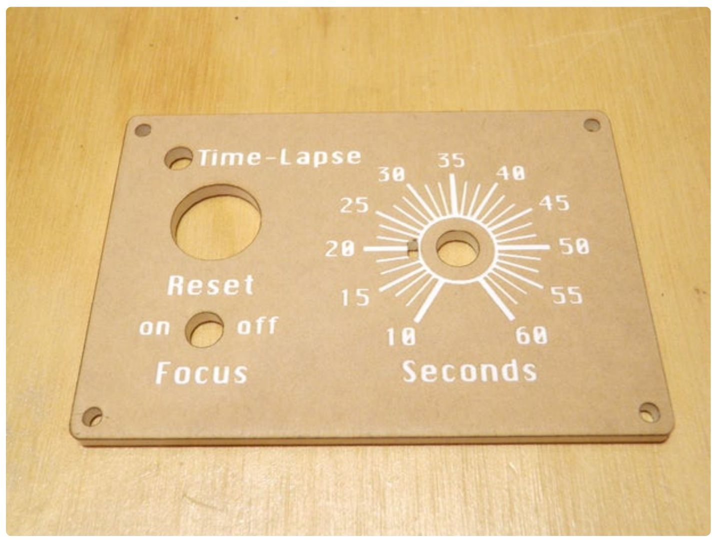
2) Screen print the design onto your surface and then cut out the template with more traditional tools.

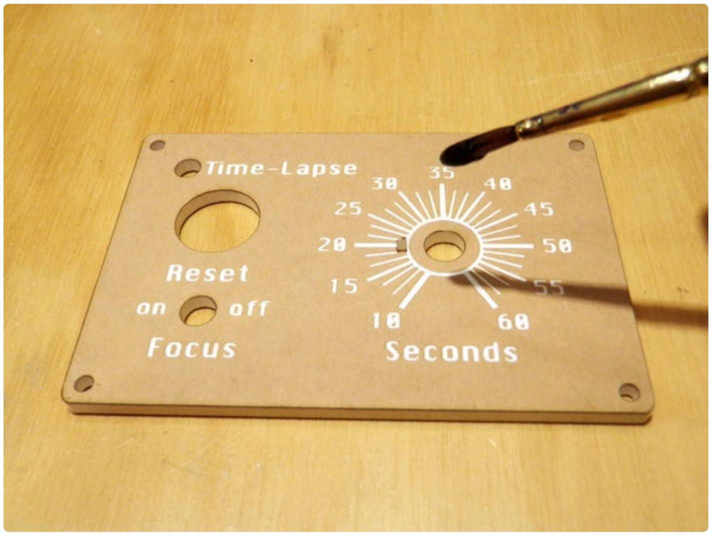
3) Download the file and use a service like Ponoko to have them laser cut it for you to your specifications.

4) Download the file. Find a local college or machine shop like TechShop that will let you rent time on a laser cutter.

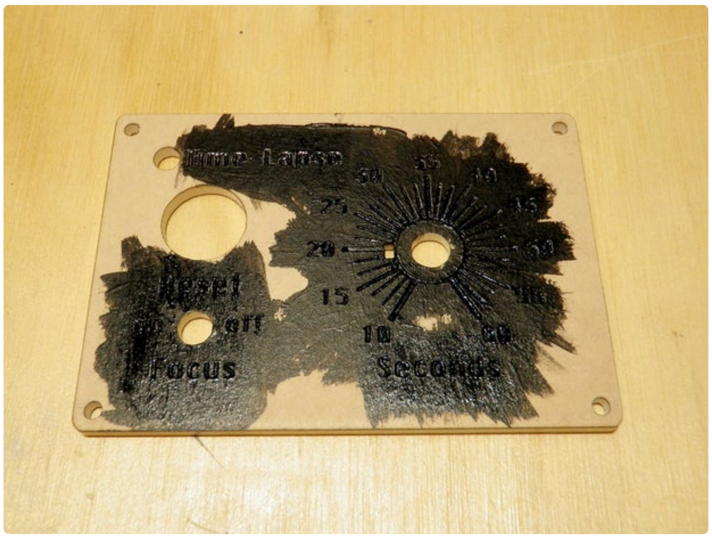
5) Download the file. Find a local hackerspace that might have a laser cutter and would let you cut the file for little or no charge.



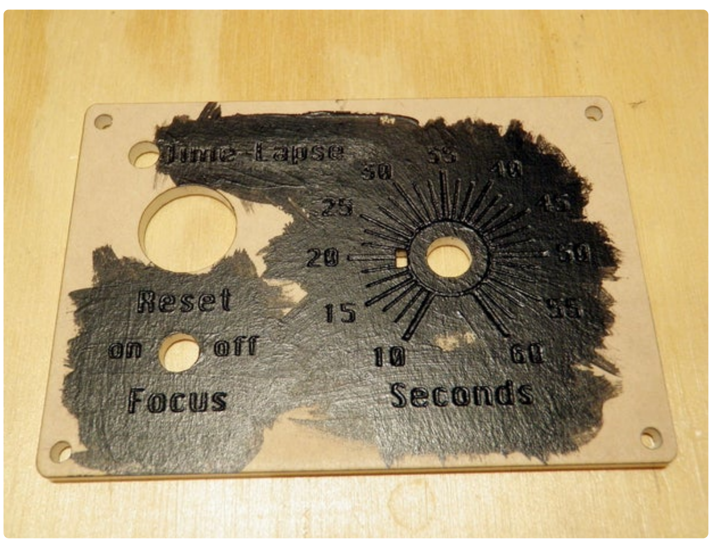


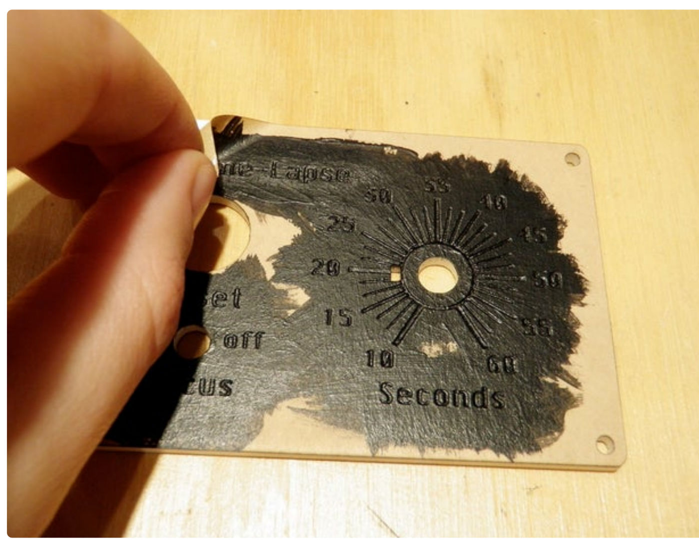

Time-Lapse
30 35 40
25 45
20 50
Reset
15 55
on off
10 60
Focus
Seconds

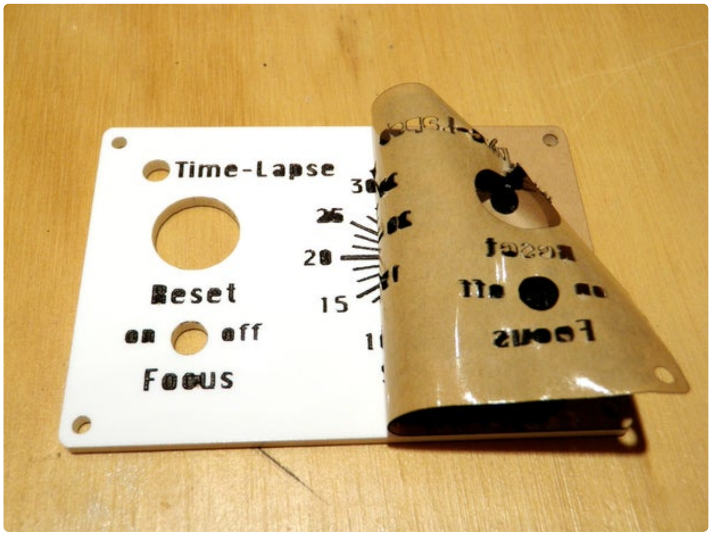

Reset
on off
Focus

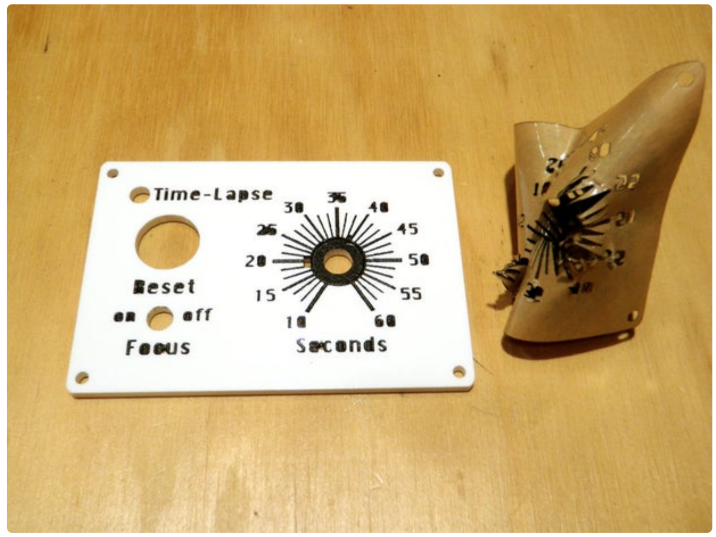

Seconds

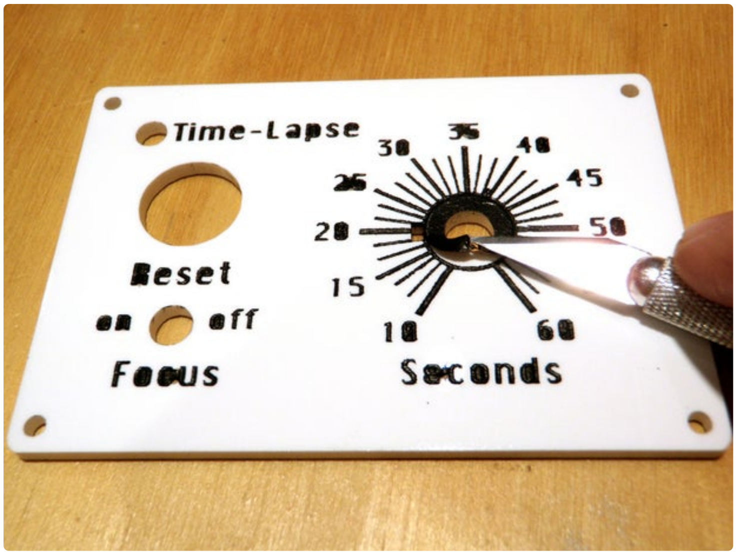

Seconds


Time-Lapse
30 35 40
25 45
20 50
Reset
15 55
on off
10 60
Focus
Seconds


30 35 40
25 45
20 50
15 55
off
10 60
Seconds







Step 3: Wire the Plug

Get a stereo cable. I got a 25' headphone extension cable from Radioshack and cut out a 4' section from the middle of the cable to use. I will use the other two remainder parts for future projects.

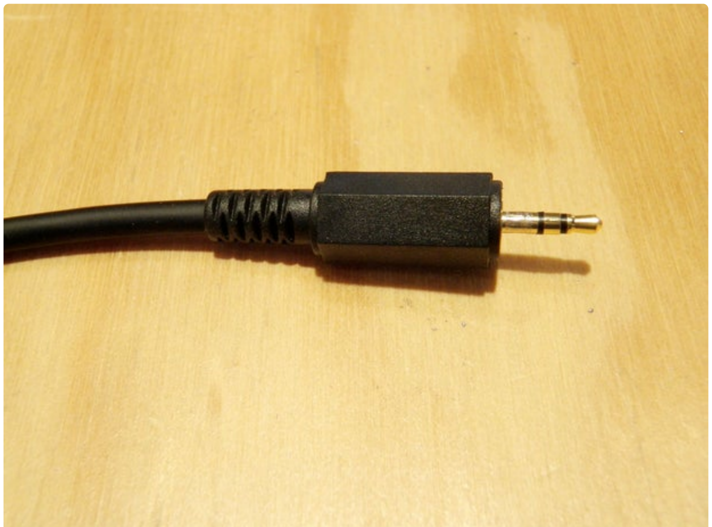
Untwist the plug so that the terminals are exposed.

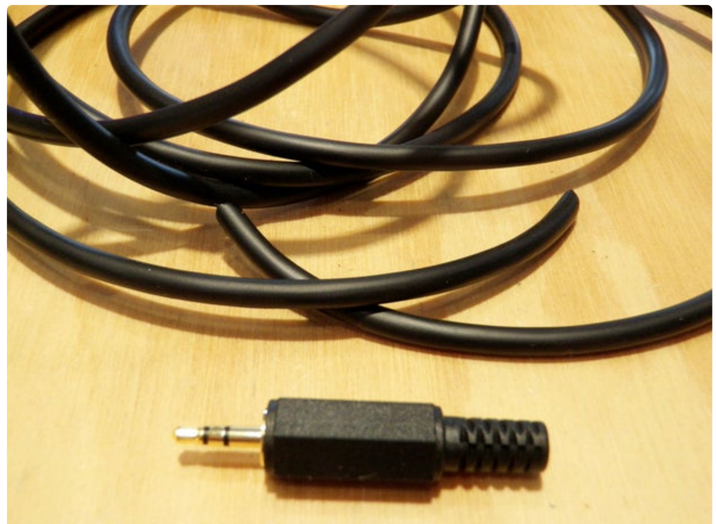
On the terminal closest to the actual 'plug part' solder the black wire from your stereo cable.

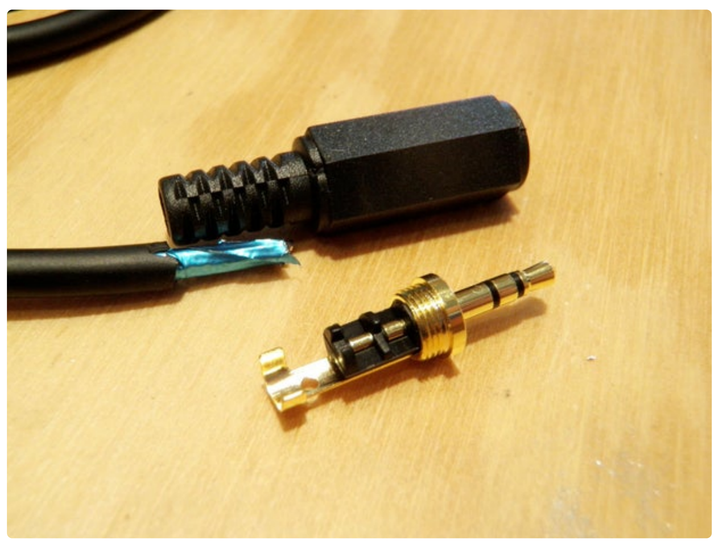
On the next terminal, solder the red wire.

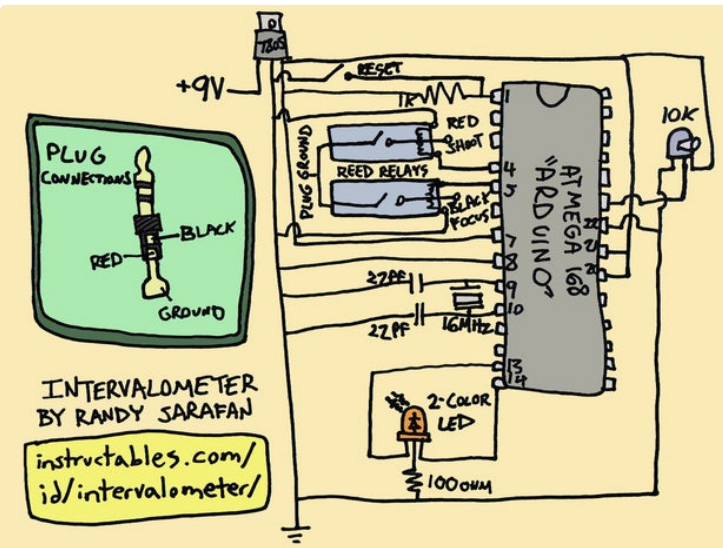
To the big metal ground tab that extends out the back, solder the ground shielding.

Check the connections with a multimeter to make certain none got crossed and then twist the cover back onto the plug.




ESD
SAFE
175D






Step 4: Drill

Lay your front panel over the opening of your box. Use the template as a guide to mark the four corners of the box for drilling. Also make two marks on the side of the case. One of these marks is for the power jack and the other mark is for the audio cable.

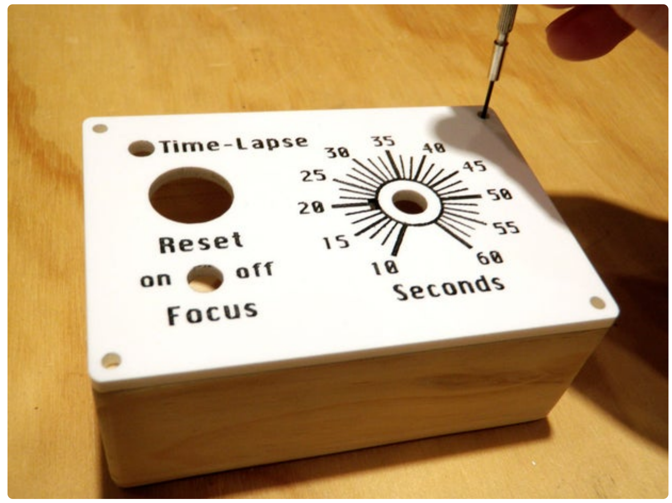
Please keep in mind that the circuit board will lay on the bottom of the case and switch and knobs will be protruding from the top. Place these holes somewhere in between.

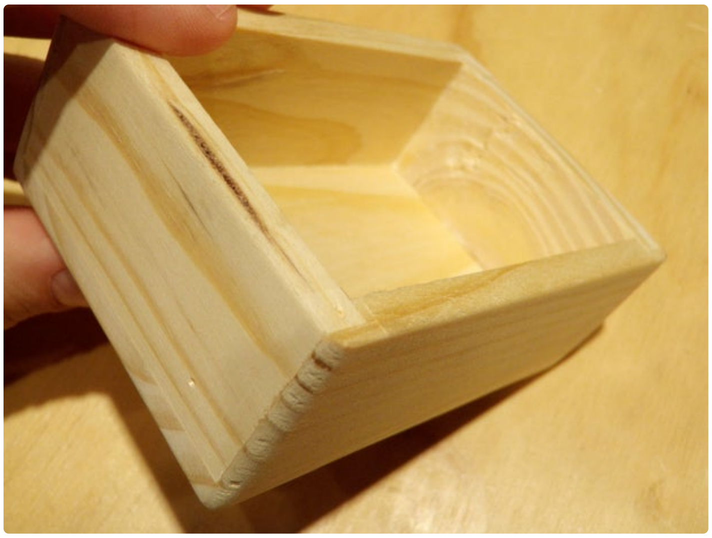
After you happy with all of your markings drill the holes.

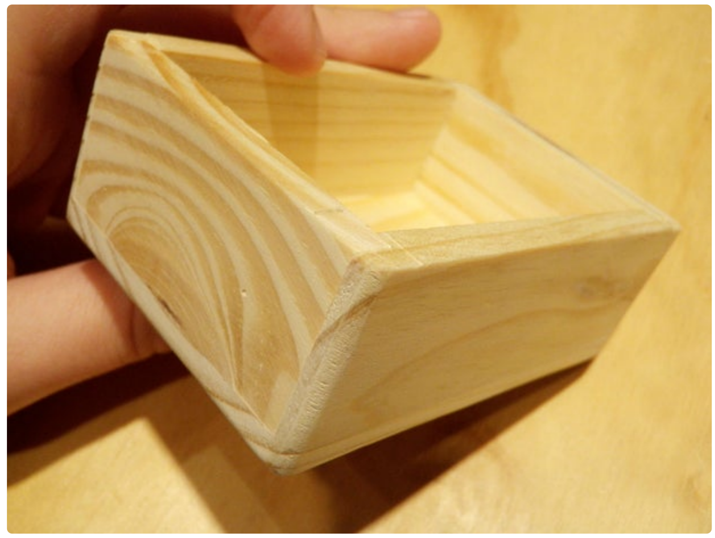
For the pilot holes for the screws I used a 5/32 drill bit.

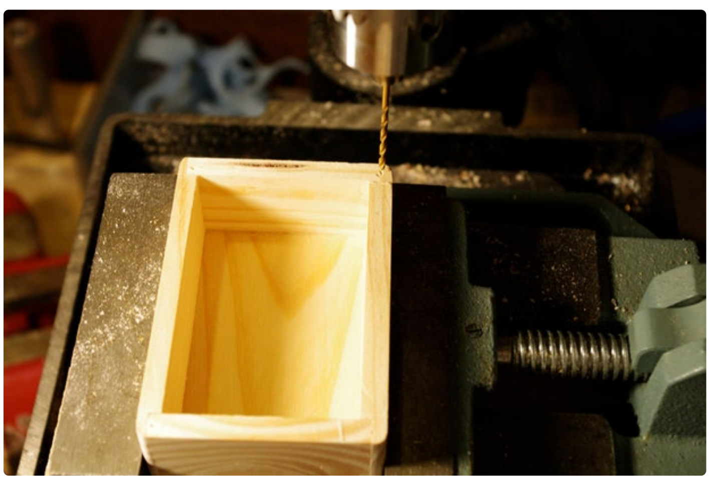
For the M-type jack I used a 3/8" bit.

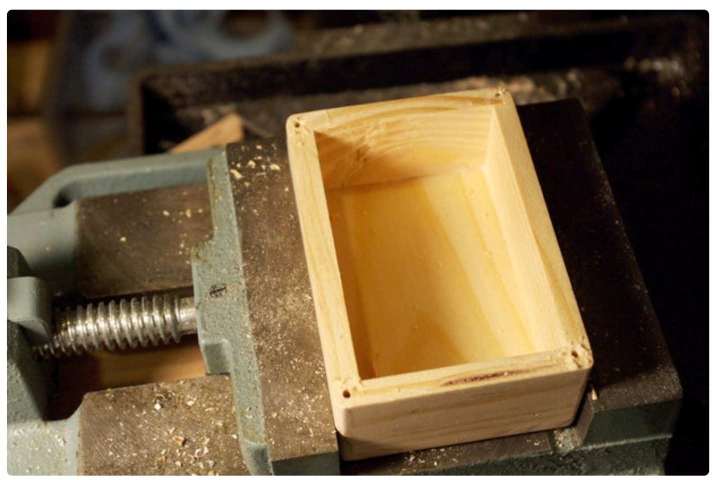
For the stereo cable, I used a 1/8" bit.











Step 5: Start the Circuit Board

Start soldering the parts onto the circuit board.

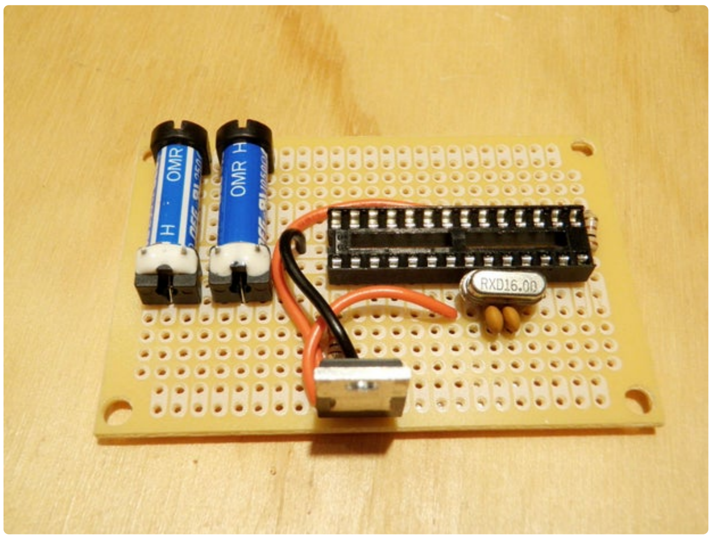
For now leave off anything that connects to the front panel and the power jack.

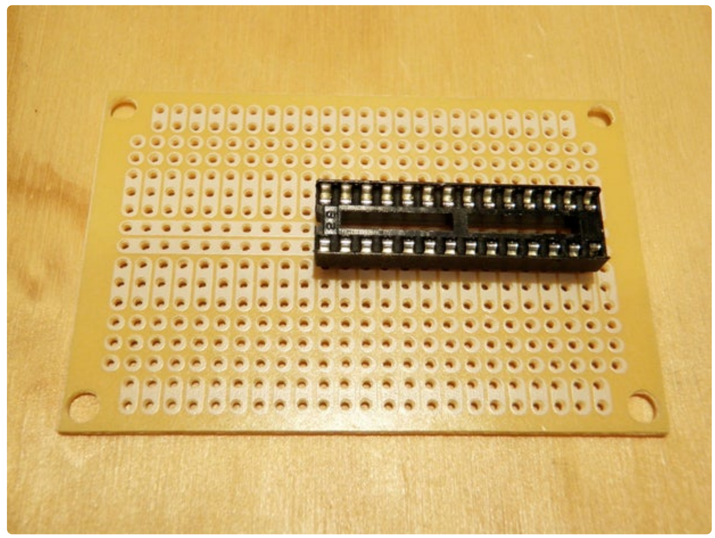
Basically, attach the parts and connections needed for the micro controller and the relays.

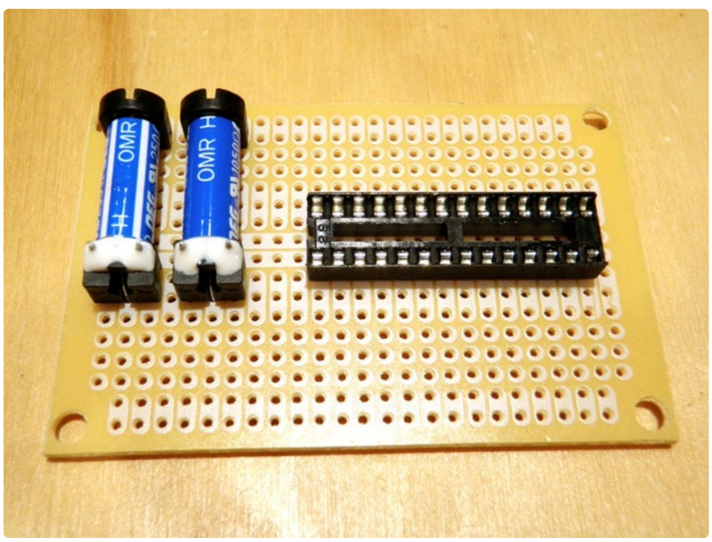
You may also want to attach a 10uF electrolytic capacitor between 9V and ground and another between 5V and ground. I thought I could get away without them, but ended up adding them because it wasn't functioning very reliably running from a 9V battery without the capacitors.

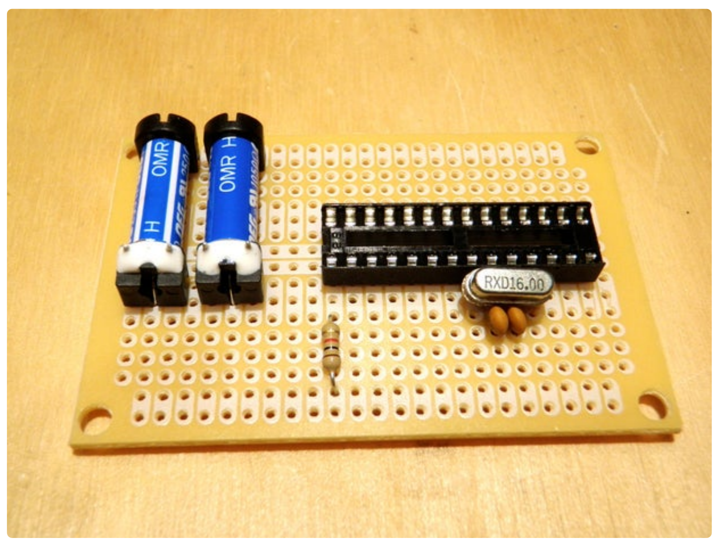
Note: in the image there are ground connections to the chip and the capacitors that you cannot see in the images since they are made underneath the board.

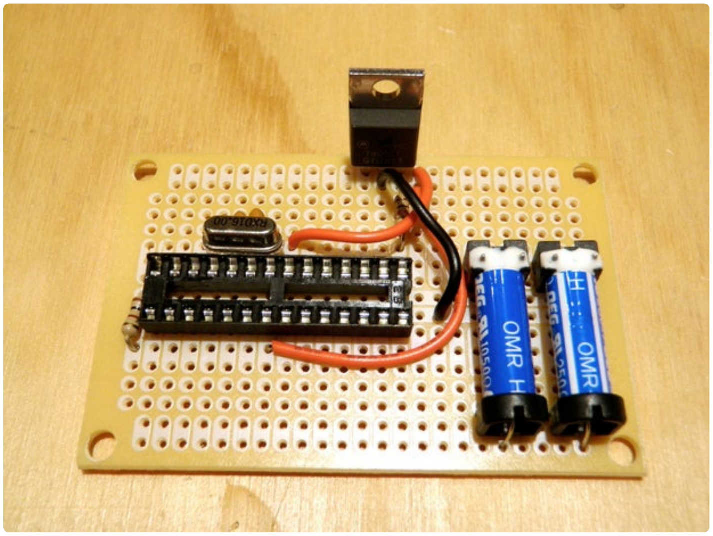
update 1-7-11: Astroboy907 converted the schematic into an Eagle schematic and board. These files are now attached to this step or can be downloaded from the comments below.

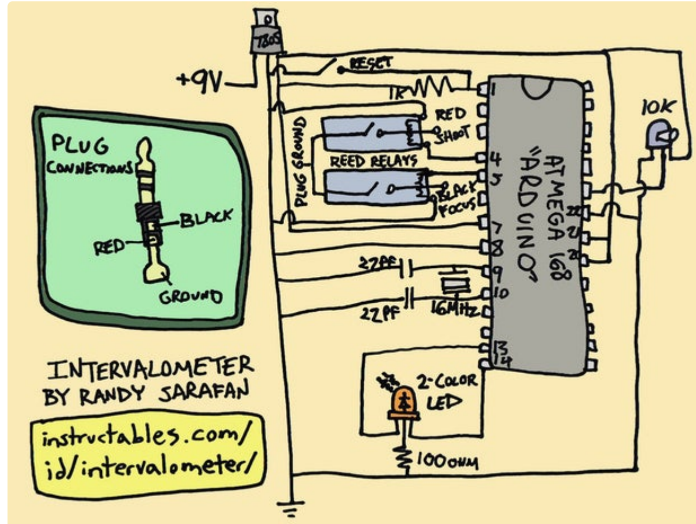












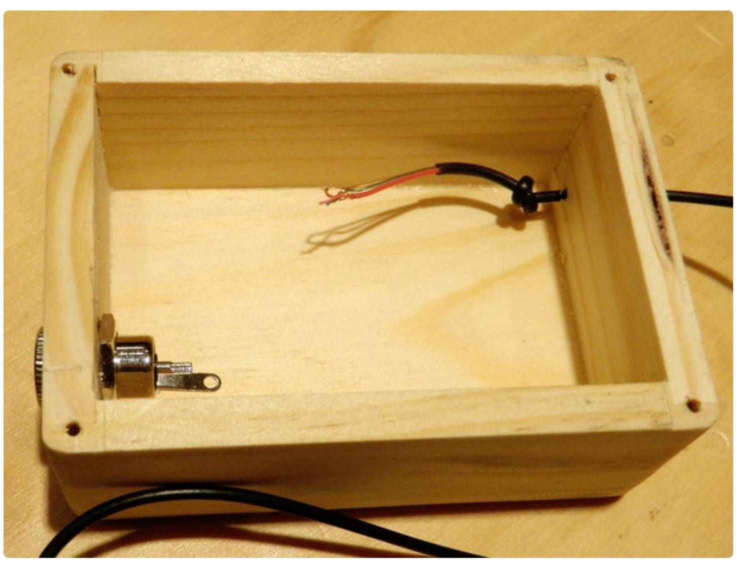
https://www.instructabl... Download

https://www.instructabl... Download

Step 6: Prep the Case

Fasten your power jack into the case.

Pass in the audio cable and tie a knot to prevent it from getting pulled out.



Step 7: Wire the Relay

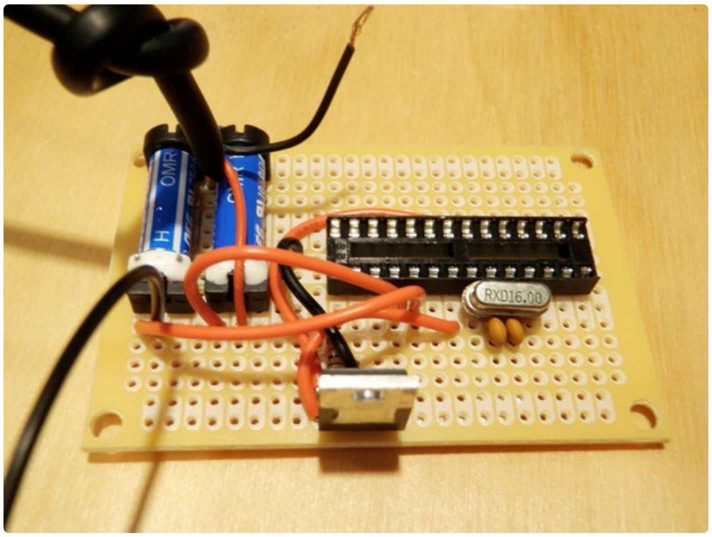
Now it is time to wire up the relays.

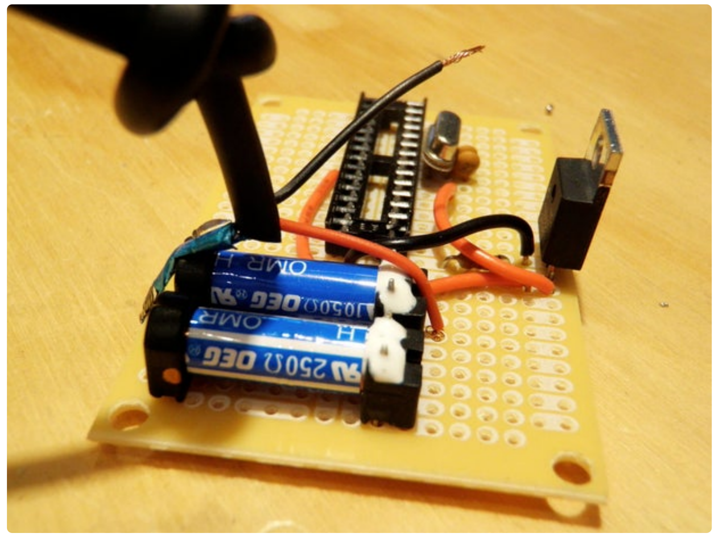
Connect one pin of each coil to ground. Connect the other pins to pin 4 and pin 5 of the Arduino respectively.

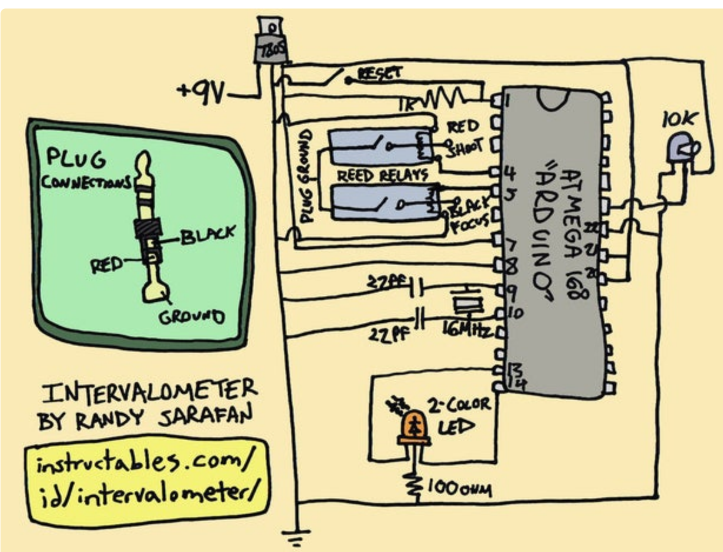
Solder together two of the relay switch terminals and solder the stereo ground shielding to them.

To the relay connected to pin 4, solder the red wire from the stereo cable to the relay switch terminal.

To the relay connect to pin 5 solder a black wire, but not the one from the stereo cable. Both the black wire connected to the relay and the black wire from the stereo cable will be solder to the focus switch shortly (not pictured in the schematic).



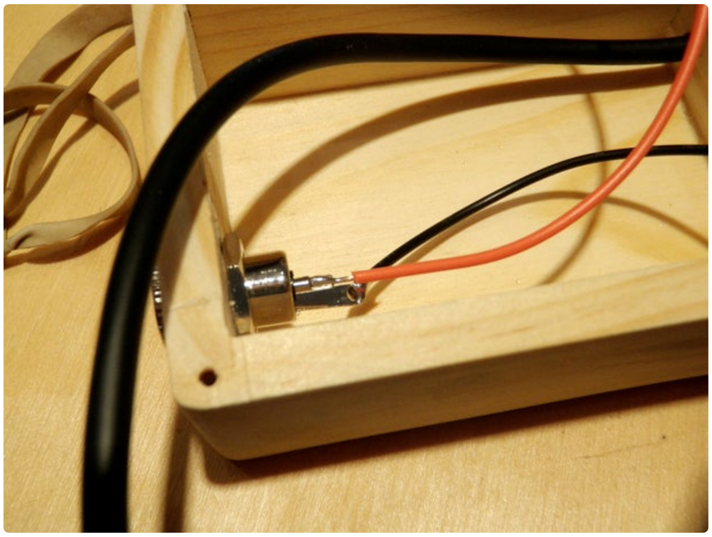


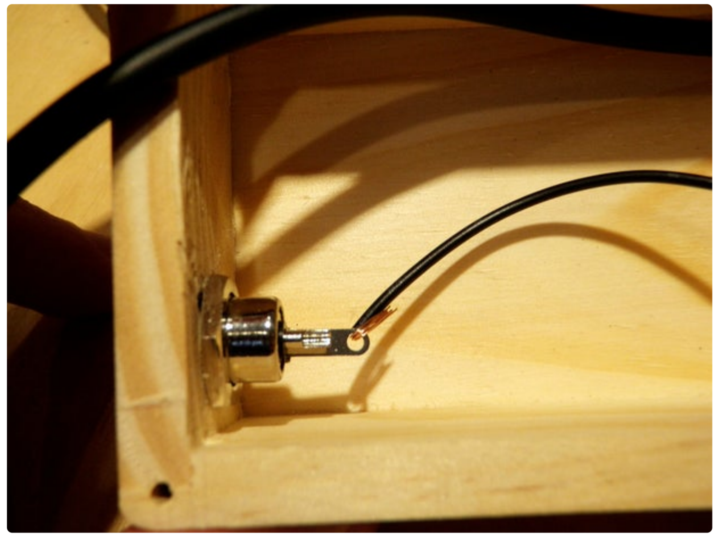


Step 8: Wire the Power

Test the center of your 6V power plug and figure out whether the center is positive or negative.

In my case, the center was positive. So, I wired the red wire to the middle pin and the black wire to the ground terminal.



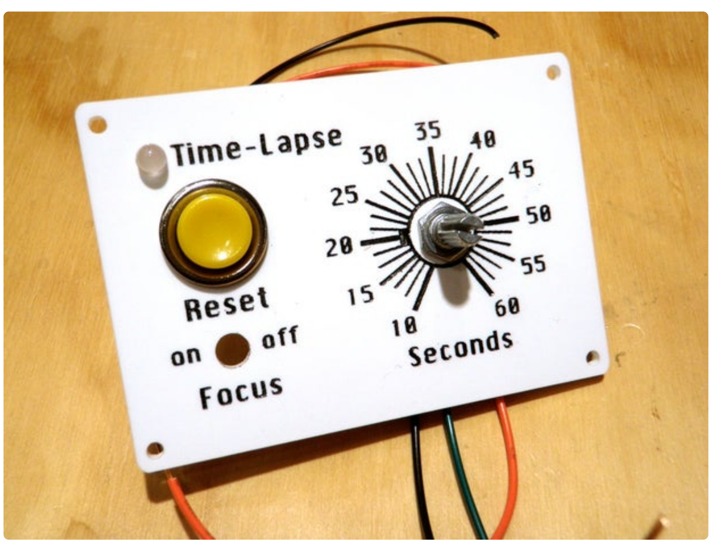


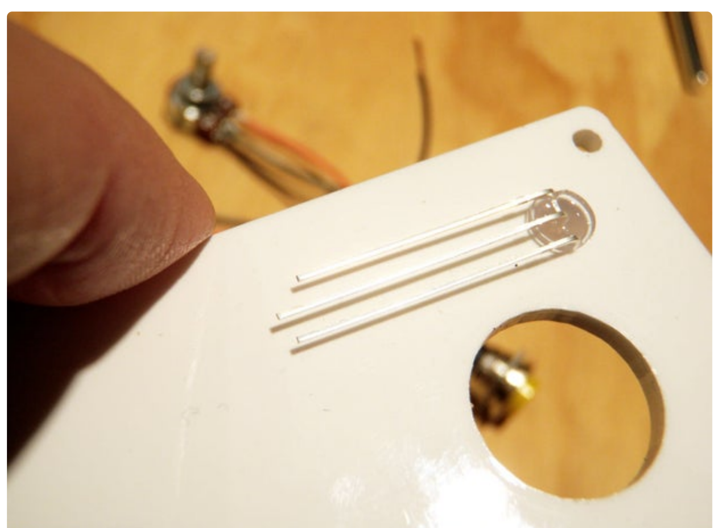
Step 9: Wire the Front Panel

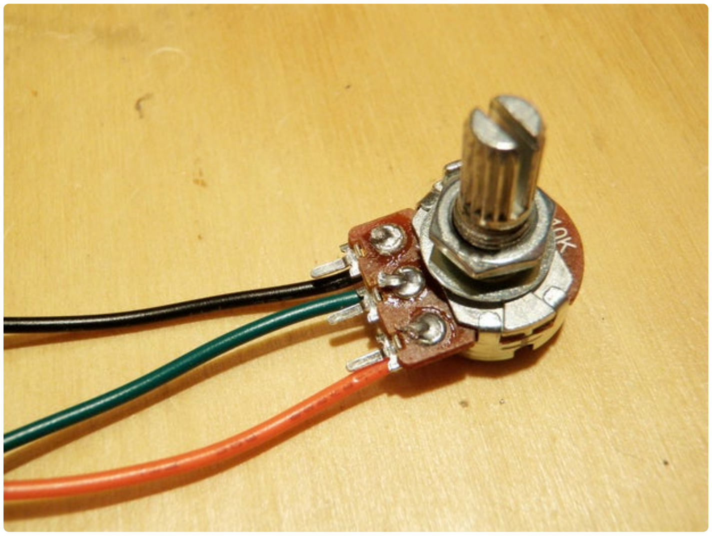
Attach a red wire to the right pin of the potentiometer, a green wire to the center pin and a black wire to the left pin.

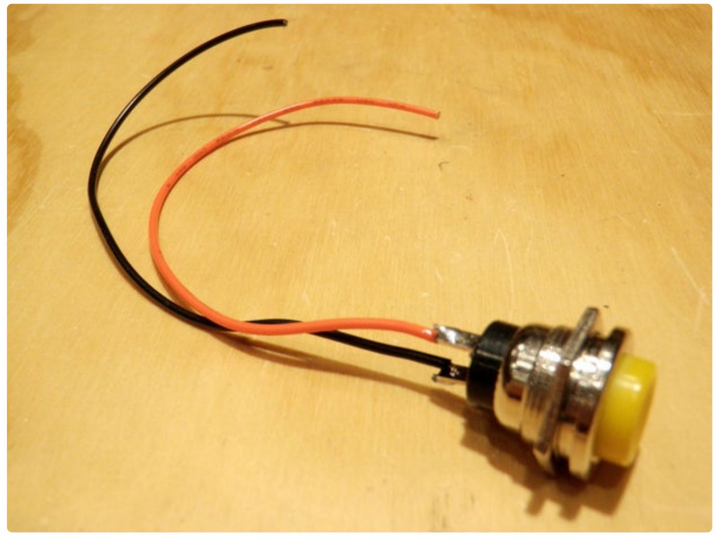
Attach a red and black wire the push-button reset switch.

Insert the two-color LED into the front panel and bend the leads to a right angle.









Step 10: Connect It All Together

Now is the confusing time to wire it all together. Of most important is the bold text. I forgot to draw this switch onto the schematic. Whoops ;-)

Connect the black wire from the stereo cable to one of the center pins on the DPDT toggle switch. Select one of the matching outer pins and solder the unconnected black wire from the relay.

Connect one of the outer legs from the 2-color LED to the other center pin (in my case this was 'yellow'). On the outer pin (adjacent to the other outer pin you have just selected), connect a wire from there to pin 14 on the Arduino.

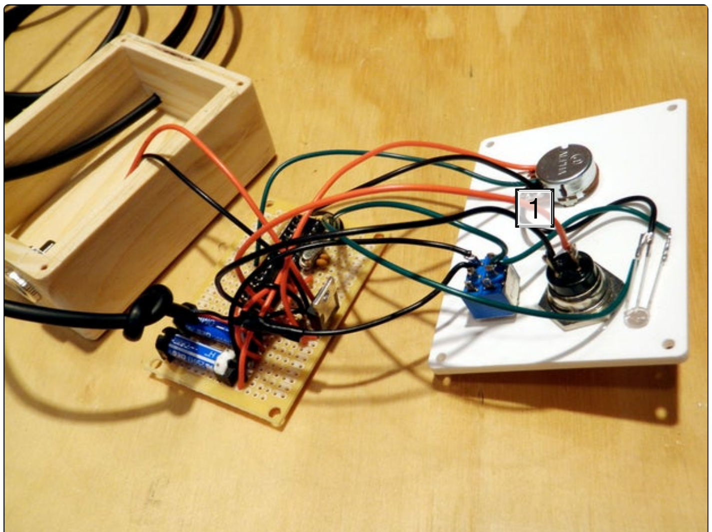
(You should be left with a pair of two adjacent unused outer pins)

Connect the center 2-Color LED pin to the 100 ohm resistor on the circuit board.
Connect the unused outer pin on the LED to pin 13 (in my case this was 'green').

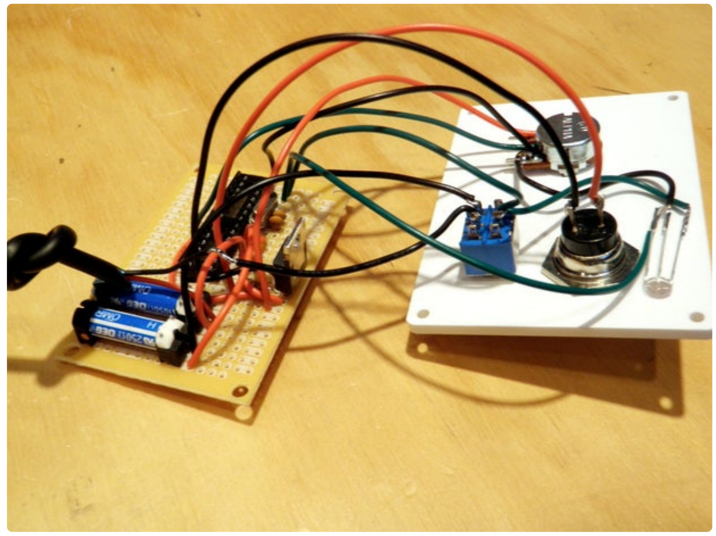
Connect the red wire from the potentiometer to the 5V power source on the board.
Connect the green wire from the potentiometer to pin 21 of the Arduino.
Connect the black wire from the potentiometer to ground.

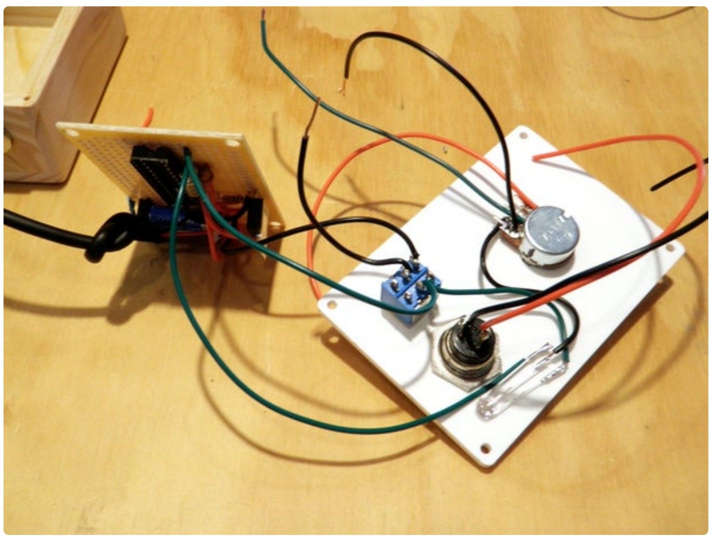
Connect the black wire from the reset switch to ground
Connect the red wire to pin 1 of the Arduino (before the resistor)

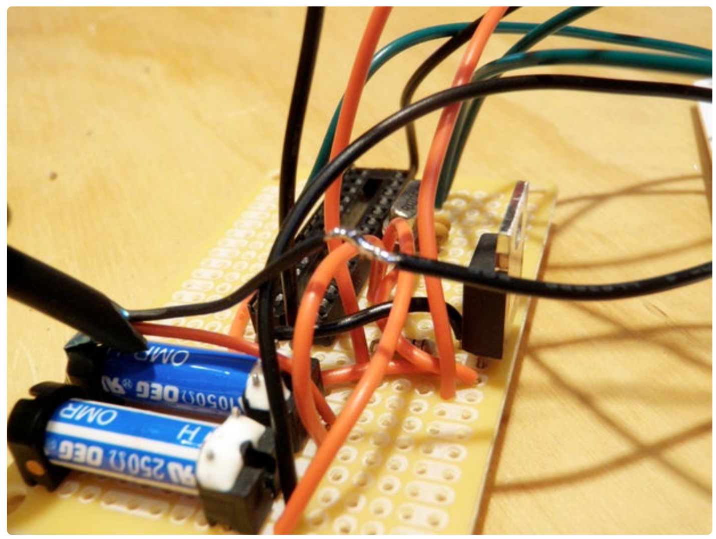
Connect the red wire from the power jack to the input of the 7805 regulator.
Connect the black wire to ground.

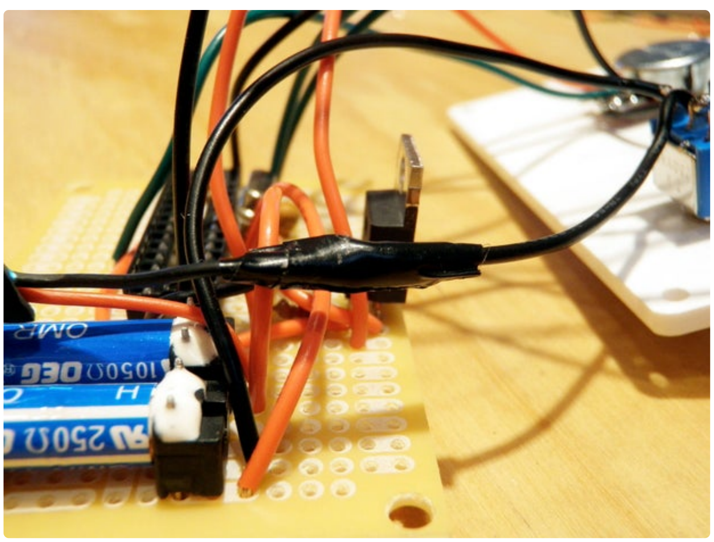


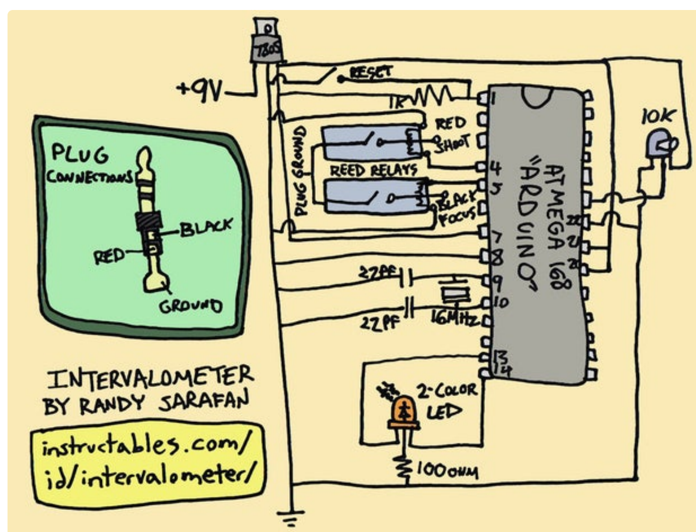
1. This black wire is wrong. The black wire should go to the 100 ohm resistor on the board and not directly to ground.









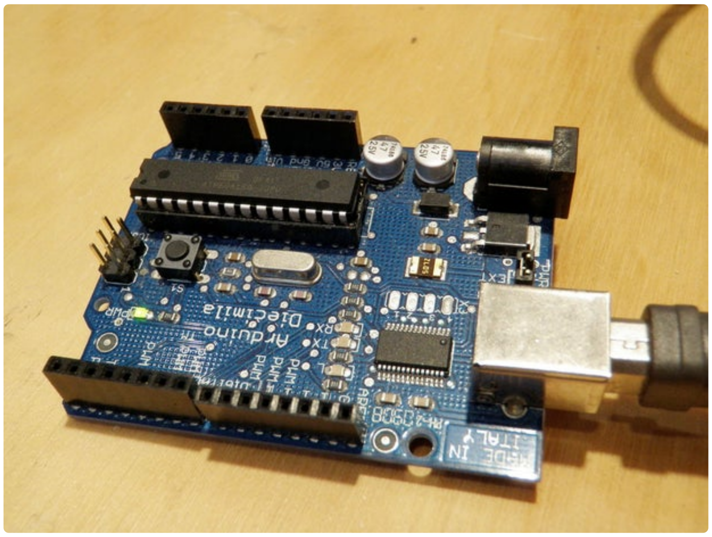


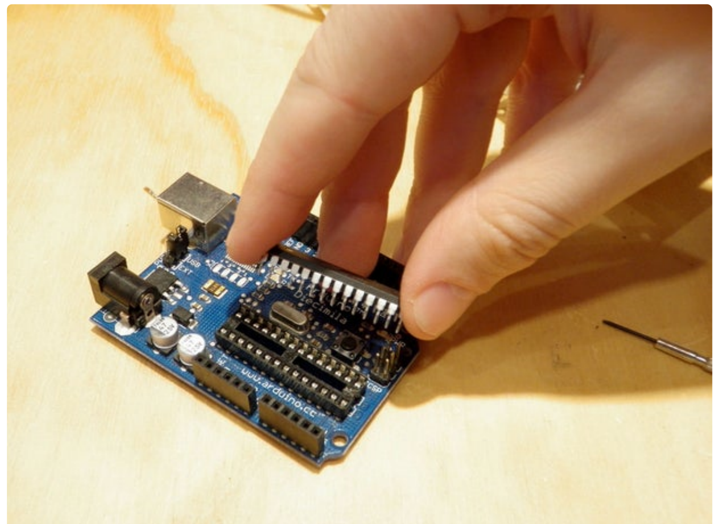
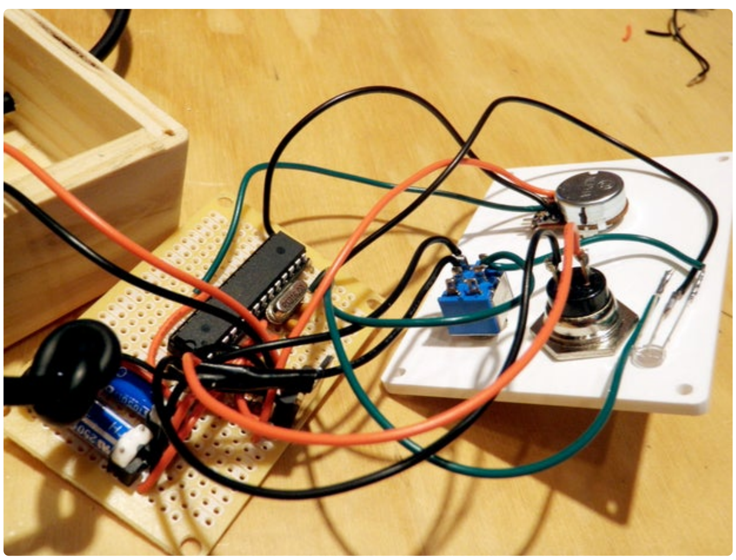
Step 11: Program the Chip

Download, compile and upload the code below onto your Arduino chip.

When you are done remove the ATMEGA168 chip from the Arduino and install it into the socket in your circuit board such that the tab on the chip lines up with the tab on the socket.

Keep in mind that your Arduino board will need a new ATMEGA168 chip with a bootloader installed on it before you can use it again.



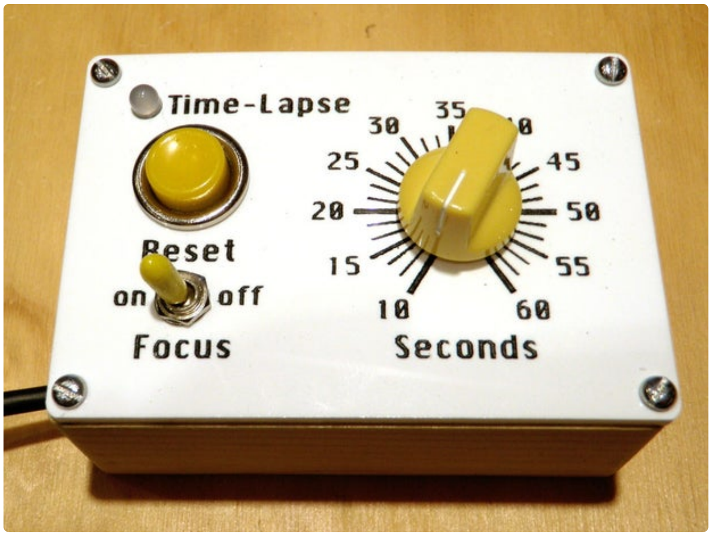
Download

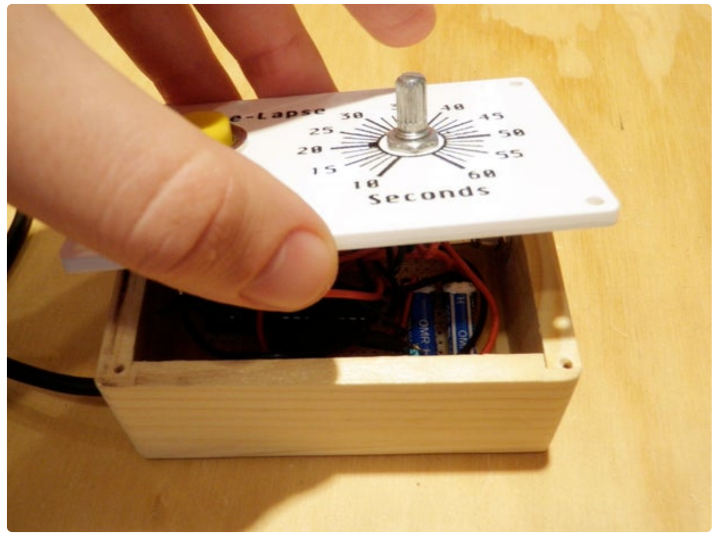
https://www.instructabl...

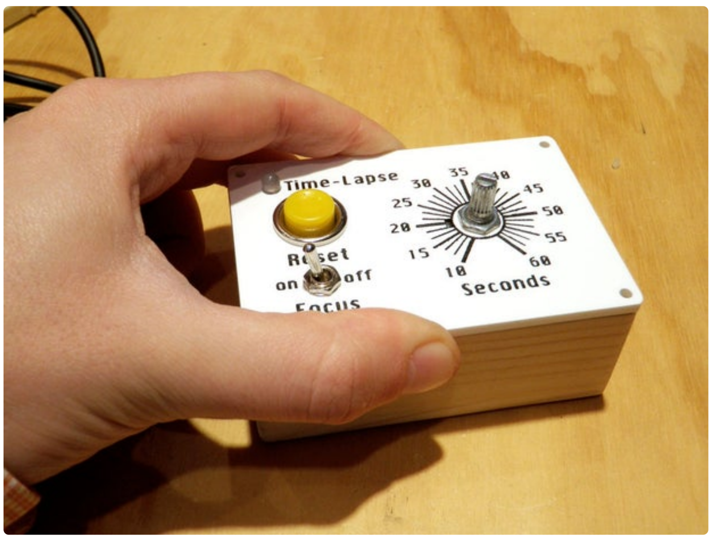
Step 12: Case Close

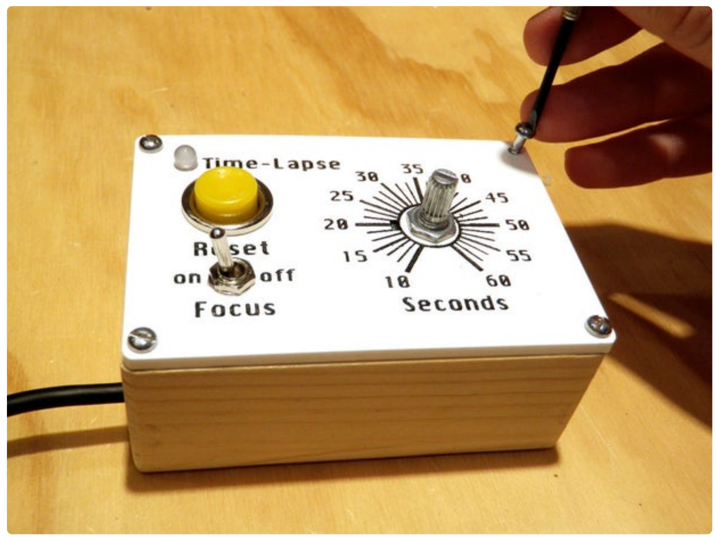
Fasten the case closed with your screws. Attach the knob to the potentiometer and stick on any toggle switch covers you might have.

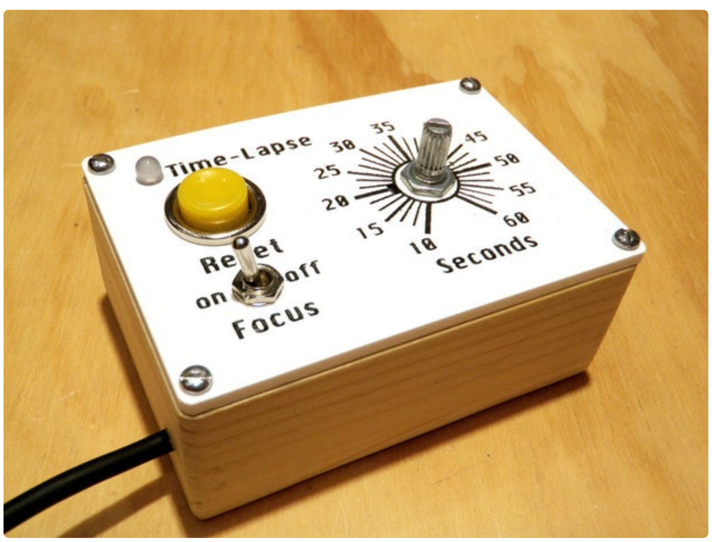
To use, simply plug it into your camera's remote control port, dial in your time settings and then attach a 6V power adapter to the intervalometer and it will start shooting away.

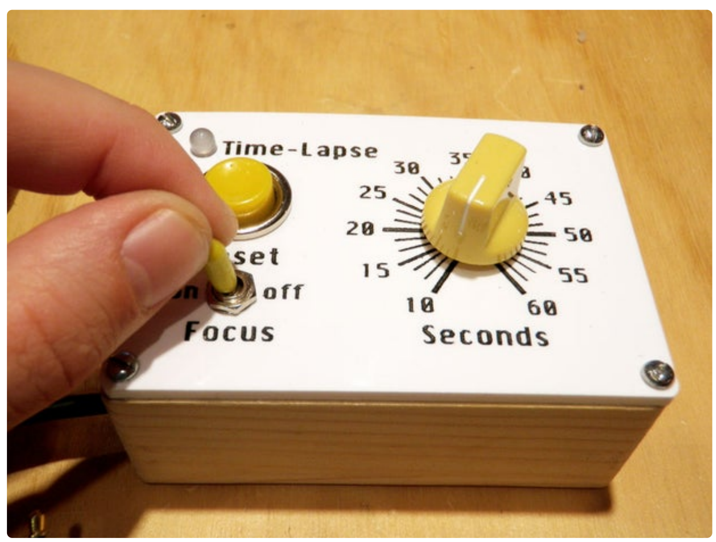












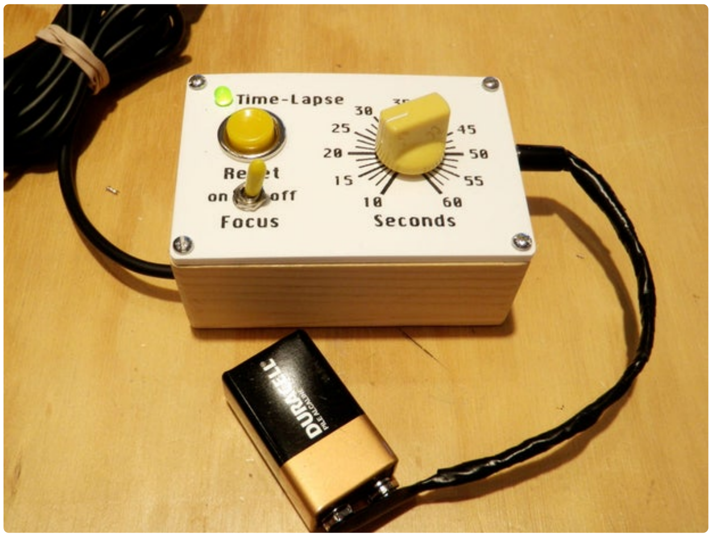
Step 13: Battery Adapter (optional)

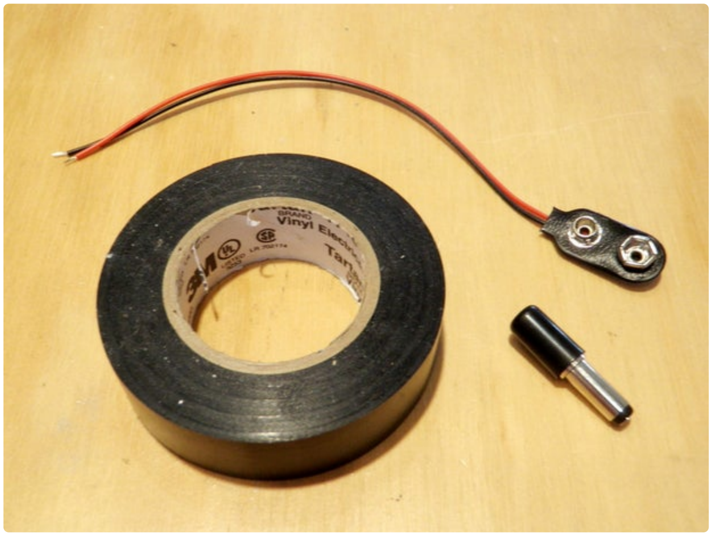
I didn't include a battery inside the case to save space and so I would not have to be bothered with a power switch or disassembling the case for battery replacement.

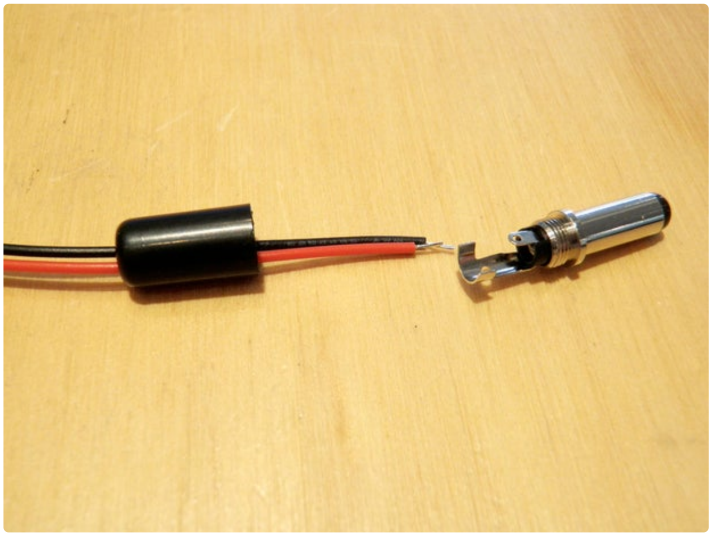
Instead, I made a simple 9V battery adapter with an M-type male plug, a 9V battery connector and some electrical tape.

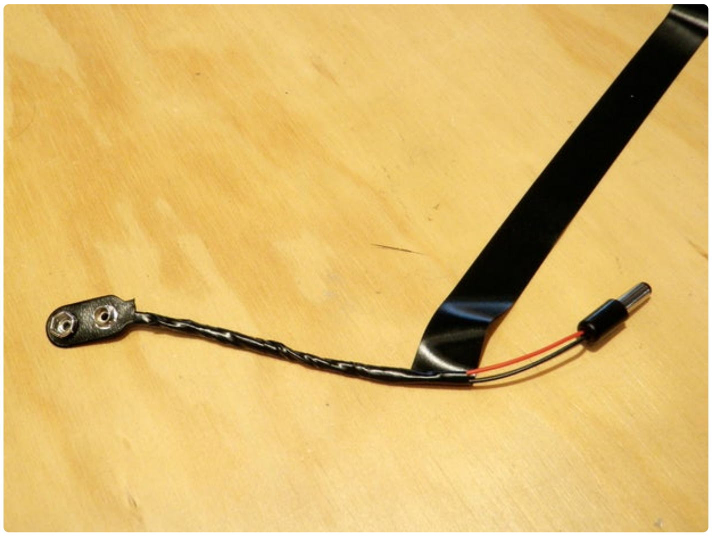
Basically, solder the red wire from the battery connector the center terminal on the M-type plug and the black wire to the large metal ground tab. Wrap the whole shebang in electrical tape when you are done to prevent it from getting ripped apart.

Did you find this useful, fun, or entertaining?
Follow **@madeineuphoria** to see my latest projects.









Hey- Heres an EagleCAD version of your schematic. Im not sure i have all my pot pins mapped correctly, and i used inductor symbols for the relay coils. I couldnt really tell where to connect the toggle, but here it is anyway... I hope it loads through attach images....

trying to upload again...

This is great! Thanks for sharing this!

I haven't even gotten to page 2, and am blown away by the quality of your control panel...even if I don't learn anything new about intervalometers (doubtful, I'm new!), I hope to pick up some design tips about UI! Beautiful!

ooh...you got me! I do appreciate that you provided more reasonable alternatives, but I was seriously hoping for a EUREKA! moment involving more-common household items....Regardless, still a good tutorial!

will this support a nikon d3200. Will it be compatible because my camera uses a usb sort of thing for controlling remoteley instead of a stereo jack.

You would have to get the proper connector for your camera. I don't know if the protocol for controlling it is the same, but it should be fairly similar.

Is that a K-3 you've got there?

Warning

Be very sure to use relays that require less... then 40Ma to turn 'm on to prevent frying your Arduino. If you can't find 'm/ want to use a normal cheap relay, you'll need to add a transistor + resistor, capable of handling more current. Google "Arduino Relay" for more info.

is there a way you could modify the circuit to increase the interval time? I've been trying to find a schematic that will go over 5+ minutes...

Hey Randofo- recently became interested in this project again. I'm trying to make a wireless remote for my FZ100 camera. I found some schematics for the remote- so thats not a problem. But I'm trying to fit it all into a 3x2x1" radioshack box. Could I use transistors rather than relays to save a bit of space? I have to fit the 9v battery, arduino (perfboarded), receiver, and camera resistors to the project- so using some 2904s would work really nice.

Also- do you think it is alright if I have the intervalometer combined into the receiver? Or should I just have an AUX input for it?

Many thanks- hope to enter my completed creation/guide into next weeks wireless contest :)
-Astroboy907

I just recently built a similar intervalometer based on a 555 for my Olympus E510 and I have successfully replaced the relay with a 2n2222 (or similar) transistor. I'm no electronics expert, but everything looks ok voltage wise and I haven't cooked anything in my camera yet.

I found some relays (eventually... my Electronics teacher had some) NAW5-K (i *think*) relays- they are about the size of three 3904 transistors back to back and run on 5v, so I was really happy with the size and they perform really well :)

I think by relay, you mean potentiometer? :)

This is beautiful!! I've made a nearly identical circuit, with Arduino, potentiometer, relays...
At the end I've permanently removed the focus (I should never use it) and I've replaced it with the option to use a photoresistor wich interrupt the cycle in the night.
I used a logarithmic potentiometer so to choose with the same precision the small intervals and the bigger one. So my scale go approximately from 6 sec, through 15, 30, 60, 120 sec. Anyway a

digital scale (also discrete) should work better, so my next project should be with a push-button and maybe a display or some leds to show the value.
Ciao :-)

I considered using a log pot, but in the end decided it would be easier to use the linear pot.

I currently let it focus for one picture, but then turn it off. In retrospect, I probably should just set it all up myself in advance. Maybe it will be useful for something sooner or later...?

You are probably right about the digital delay :-)

Since it's all controlled by an arduino anyway, couldn't you just take the log of the input from the linear pot in software? That seems like a better solution anyway - less soldering if you want to change the behavior :)

This is true. Will keep in mind for possible future version.

great! here is the first time-lapse I've made with my circuit (and an old Panasonic FZ20)
http://vimeo.com/17430724
Remember that the mirror in SLRs is damaged after a lot of shots, usually 100.000 I remember, you reach that in a week with 10 shots per minute..
bye!

Most good quality SLR's have a mirror 'lock up' capability. ..which also makes it much more quiet.

Oh, it's right, I've completely forgot that... I've to try with my 40D. Thanks!

I have GOT to get on board one of these days with my own Arduino and a decent book on the subject (for a noob like me as I'm used to building stuff like this with timers /555-556/, sequencial counters, flipflops, logic gates, etc.) A lot of wiring, a lot of room for mistakes.

Take a look at Nerdkits.com. I've been using them for two years - very happy. I've learned to program Atmega chips and built some nice projects.

Looks nice...I will have to look at it closer when I "get home" :-)

I came the same route as you - gobs of individual ICs wired together, with my bibles "TTL Cookbook" and "CMOS Cookbook" dangling off the edge of the desk. Voltage regulators running so hot that they would burn the skin (even with the big heat sinks). I took a microcontroller course 10 years ago, but never used it.
Two years ago I stumbled onto Nerdkits - a couple of MIT grads put the company together. For around $80 they will send you a serious Atmega microcontroller, programming environment for Mac and PC, USB cable, breadboard, 4-line LCD display, temperature sensor, speaker, switches, wire, leds and a CD full of theory and excellent example programs.
You will have to learn C language, but that's not so bad (I come from FORTRAN, PL1, BASIC, PASCAL, and assembly - C is better). You will also have to learn the logic of bitwise manipulation, so that you can change one bit of a byte-wide port - pretty easy once you grasp it.
What you get for your money is a complete, stand-alone industrial microcontroller with it's programming environment, plus lots of examples of how to put it to use.
Nerdkits site has continuous forums on anything you could want, with real newbies asking, and getting answers to all of their problems. The two owners of the site also respond in the forums, and even respond to email questions (I don't know how they handle the volume - but they do, quickly and with friendly help).
I've built several real-time clocks with different displays. One is a "one digit clock" - it uses one 7-segment LED to tell the time: at 2:35 it flashes the "2", then the "3" then the "5". Under the display is four regular leds - each one lights to let you know what number you're looking at: the 2nd one lights when "2" is displayed, the 3rd one lights when the "3" is displayed, etc. I love this project - I had to learn how to tell time with the microcontroller, and found that it wasn't accurate. Then I

bought a second Real Time Clock chip with it's own crystal and battery backup, and had to learn how to interface that with the MCU using the MCU's serial port. (It has several - different modes, several analog-to-digital converters, etc, etc, etc). My other hobby is woodworking, so I built really nice cases for the clocks using spalted and curly woods. My parents and girlfriend each have one and love them. Of course, I don't have one for myself (of course), but I'm building another one very soon, using a larger, 2-1/2" LED, which, of course, needs higher voltage than the regular LED (which the MCU can drive itself) so now I'll have to learn how to switch higher voltages (8 volts) using transistors triggerd by the MCU's 5v outputs. It's been a wonderful learning experience. Sorry for waffling on so much, but I think you owe it to yourself to buy a kit from Nerdkits - you'll curse me when you're learning the ins and outs (the included datasheet for the MCU is about 400 pages of fine print), but you'll thank me forever after you get over the learning curve. Their examples will have you up and running your first successful project within an hour - a digital thermometer that displays temperature on the LCD and on your computer's serial port. You'll thank me. Go buy a Nerdkit- no doubt about it. (No, I am not affiliated with the company - just a fumbling hacker who lucked onto their site.)

It sounds intriguing, but the $80 price tag will delay me for a bit. Probably a few months at the very least. :-)

Thanks for this very useful tip. It is this helpful exchange of information with no profit motivation that enriches our lives and makes Instructables so nice. David

theres a boatload of software for PCs. for mac ive had good results with time lapse assembler :)

there might even be a pc version...

The files are showing up as tmp files. Doesn't seem right as they should be sch or brd files. What way do i open them

If it is doing that, just rename them to the file name shown here once they are downloaded.

also be noted that the parts are not to scale or are just fakes... youre not going to get anywhere making an actual board off this without updating parts and/or making your own part libraries. I just finished my own version and added some different code and a pushbutton. The codes a bit buggy though so any help with it would be appreciated. Feel free to use this code (i did update it so it would work on an actual board, not the hackduino used in the actual project). I got it to run 10-60 seconds and an alternate 1-6 min mode. Thanks! Great project :)

Nice, a 555 timer would have done the trick as well.
Just wondering why you had it start at 10 seconds, I found 4 to be good for clouds and other relatively fast moving objects?

A standard 555 timer is not a good substitute. Standard 555 timer chips have start up delay inaccuracies. The issue is well documented. There are a number of manufacturers that make a cmos version specifically for critical timing applications. Depending where you live they can be hard to obtain and expensive.

I'm aware of the inaccuracy, but for this that would be of no consequence.
An Arduino is easier but more expensive.

The Arduino was simpler with the auto-focus and required fewer external parts.

It seemed like a good idea at the time. I figured that I probably wouldn't want to shoot anything faster than that.

Nice idea.

Just a quick easier trick for the decal. Anyone that etches their own printed circuit boards should know this.

- Just print it in reverse on Bubble jet photo paper but on a laser printer.
- Place on the backing and iron on with an old clothes iron.
- Let cool.
- Soak in hot water for about 10 minutes till cooled down.
- While still wet paper will just about slide off. Rub slightly for remaining paper.

Presto label applied and ready

This is very interesting and regretably electronically beyond my abilities. Recently there was a mechanical Instructable which I could probably manage, but my question is, how does one convert a series of still images to a video? I hope someone can tell me.

do a google search for time lapse software, there are many free programs to create video from all your still images.

If you are lucky enough to have access to a Mac, simply open QuickTime Player 7, and choose "File --> Open Image Sequence..." and it will open all the still images in the folder you select and turn them into a movie.

Here's a sample from my Olympus OM-20, which had a built-in intervalometer that I sorely miss.

(It's a 142MB download. Be patient.)

Thanks for the suggestion but firstly I don't have a Mac and secondly my internet connection is very slow - certainly too slow for 142 MB. Another suggestion from another helpful instructable-er is to use a free download program http://www.ndrw.co.uk/free/jpgvideo/index.html. bfn, David

@randofo
you might instead use a saw for the frontplate.
and use a piece of pcb instead of acryllic-glass.
in that case you could print the layout and etch it on the pcb.
this is eassier to do at home. you only need some sodium hydroxide, hydrogen peroxide and hydrochloric acid. and prefereble an laserprinter for the print. dont forget to use parchment paper(im not shure this is the right translation for what i mean. i am refering to a kind of paper wich is fat-proof. a litteral translation would be fat-free paper.) instead of normal paper.

he MIGHT have done a lot of different things, but the average person CAN'T do their own circuit etching, as they have not the know how... AND while it is easy... MOST don't feel comfortable with the overly technical things...

I can say, in response to this kind of comment... go make your own and do it that way...

Heck... he COULD have just bought one

Beautiful job! I like it... It looks professional and It's a good addition to anyone's camera bag... 5 stars!

With just a little smushing it fits fine on the front of an Altoids tin. Now i gotta eat all the mints to see if my arduino will fit inside! :)

I think it will fit! Now everything i see almost could be turned into an enclosure :)
Other on the cheap solutions- soap box, old (minidv?) tape box, the box the arduino came in, dollar store boxes... cardboard boxes. a lot of solutions if you dont happen to have a laser cutter or a

fancy enclosure near by :)

Hey- I received an arduino uno for Christmas (yay!) and was wondering if i could do this without the crystals and 22pf caps. I have everything else needed for this project (except my 5v relays are not reed relays). Could I probably edit the code to have a wider range of time? Anyway, great project and just what i was looking for for an arduino intervalometer.

I'm used to putting the crystal in because that is what I've always done when working with micro controllers. A friend of mine told me that was completely unnecessary with the ATMEGA168. I kind of don't want to believe them, but I imagine they might be right that it is built into the chip. You can try and if it doesn't work, the worse that will happen is that you will need to add one later.

If you remove the ATMEGA168 from the Arduino, you will need to get another one with the bootloader already installed before you can use the Arduino again (or get a new one and burn the bootloader yourself).

yes i was just thinking of including the entire board for now.. its an ATMega 328 i think, so i should be good, just finding the chips around here is a bit of a pain... i usually end up ordering online parts cause the only thing near is radioshack

Join the club... :-)