

Build Your Own Arduino Pulse Sensor

Visualize Your Heartbeat DIY

By Jason Tung



Description: Arduino Pulse Sensor

Assembly Time: 5+ hours

Skill Level: Intermediate

I've always been fascinated with technology's role in medicine. We live in a world where technology and medicine are intertwined, and I wanted to try my hand at a DIY medical electronics project that could do just that. Our friends at Make had built an [Arduino heartrate monitor](#) that displays the results on my computer just like it would in a hospital. My interest was piqued, so I decided to give it a try.

The finished project fits over a fingertip and works by measuring the amount of infrared light reflected by the blood circulating within. Of course you can take your own pulse rate using your fingers and a watch (stethoscope optional), but this gadget will allow you to record the results and see them on the screen.

I'm an electronics beginner and found this project to be somewhat of a challenge, but I was ultimately successful and learned a lot about troubleshooting, soldering and more.

Required Tools and Parts:

[Soldering Iron](#)

[Solder](#)

[Wire Stripper](#)

[Wire Cutter](#)

[Arduino Uno](#)

[USB cable](#)

The Kit Includes:

[IC, LM324N, quad channel Op Amp, DIP-14](#)

[Transistor, 2N3904, general purpose NPN](#)

[0.1 \$\mu\$ F 50V ceramic disc capacitor](#)

[1 \$\mu\$ F 35V tantalum capacitor](#)

[1/4 watt resistor, 470K \$\Omega\$](#)

[1/4 watt resistor, 68K \$\Omega\$](#)

[1/4 watt resistor, 39K \$\Omega\$](#)

[1/4 watt resistor, 8.2K \$\Omega\$](#)

[1/4 watt resistor, 1K \$\Omega\$](#)

[1/4 watt resistor, 1.8K \$\Omega\$](#)

[1/4 watt resistor, 220 \$\Omega\$](#)

[Infrared emitter LED, 880nm, 5mm](#)

[Infrared phototransistor, 5mm](#)

[Prototyping circuit board, 1.6" x 2.7"](#)

[25' green solid-core 22 AWG hook-up wire](#)

Purchase the [Arduino Pulse Sensor kit](#).

Step 1: Gather parts and materials

Grab your kit and gather up the remaining tools and materials. Make sure you have all necessary parts and tools.

Step 2: Assemble the Board

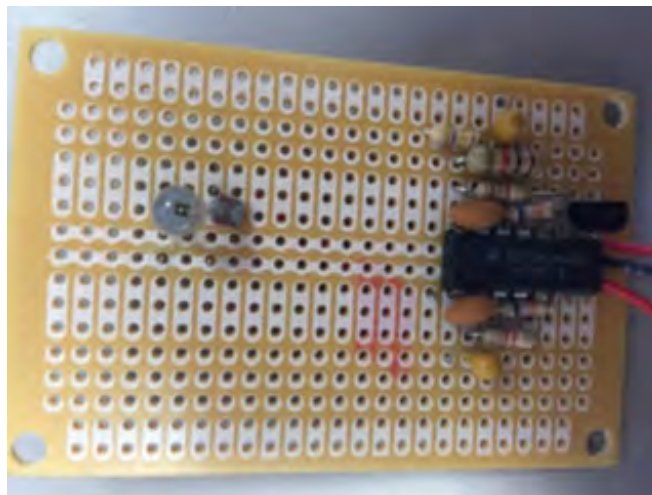
I started by adding the LED and LDR to the board. On the solder-side of the board, I placed a 220 Ohm resistor on the LED and a 39k resistor on the LDR. Before placing the LM324 chip, I clipped off leads 5-7 and 8-10 because they are unnecessary for the circuit. Be sure to put the chip one row from the end of the board, with the pins straddling the ground and power rail. I made power and ground connections by flowing solder from pins 4 and 11 to the adjacent rail. After the chip was soldered in, I placed the disc capacitors: one across pins 1 and 2 and the other across 13 and 14. I then soldered the six remaining resistors into place. The leads on both sides of the chip should be electrically connected via solder. Next up, I soldered the negative leads of the tantalum capacitors (leave the positive leads unsoldered for now). Be sure to check the diagram and make sure the positive and negative leads are in the correct location on the board.

Step 3: Connecting the Signal Pathway

I cut two small wires (approximately 1 and 1.5") and connected the shorter wire from the primary output to the secondary-stage input. The longer wire connected the LDR to the primary op-amp input. At one end, both wires are connected to the tantalum capacitor's positive lead. Then, I added the NPN transistor. Be sure to check its orientation: the emitter lead is at the edge of the board and the 1.8K resistor is connected to the middle lead.

Next, I cut a very small wire. I connected the wire from the transistor collector to the power rail (be sure to skip the ground rail). Then, I soldered another piece of wire from the primary-stage resistors to the ground rail (be sure to skip the power rail). Even with around 25 hours of soldering under my belt, I still found connecting these tiny wires the most difficult step of the whole project.

My first piece of advice for this step is to cut the wire a bit longer and fold it into a "U" shape. I found that following this step made the wire less likely to fall out of place and much easier to solder. Next, be sure to have a [desoldering](#) mechanism close by; the small wire bits easily burn.



Step 4: Connect the Cables

After the board was ready to go, I cut three wires, around 6" each. I used one wire for power, one for ground, and another for signal. I stripped about 3/8" of insulation on each wire. Be sure the wires are tightly soldered in to avoid the hassle of soldering again.

Step 5: Connect the PC Board to Arduino

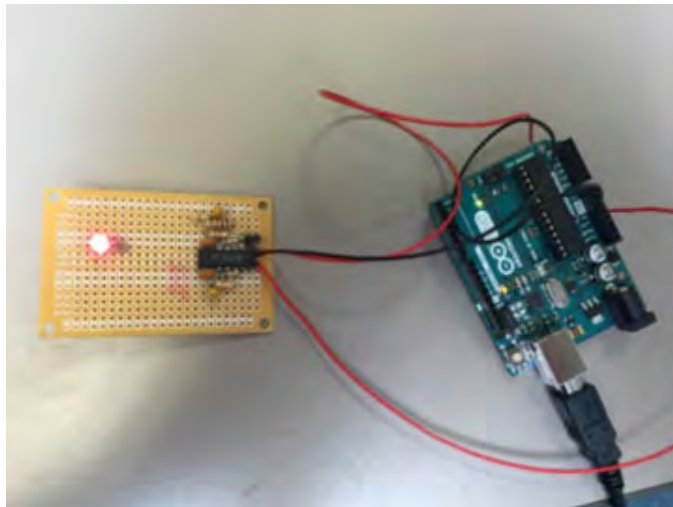
To power your Arduino, connect the USB A/B cable from your computer to the USB B port on the Arduino. The green "ON" LED on the board should light up.

After the Arduino is powered, strip around 3/8" of insulation off of each wire. Connect the power wire to the 5V pin header on the Arduino. Then, connect the ground wire to the Arduino's GND pin header. The LED should light up right away if the board is correctly connected. If you're struggling to make the correct connection, ask yourself:

- Is everything soldered in the right place?
- Is solder flowing where it should be?
- Is everything connected properly?

Once your board is successfully powered, place the wire in the Arduino "A0" pin header.

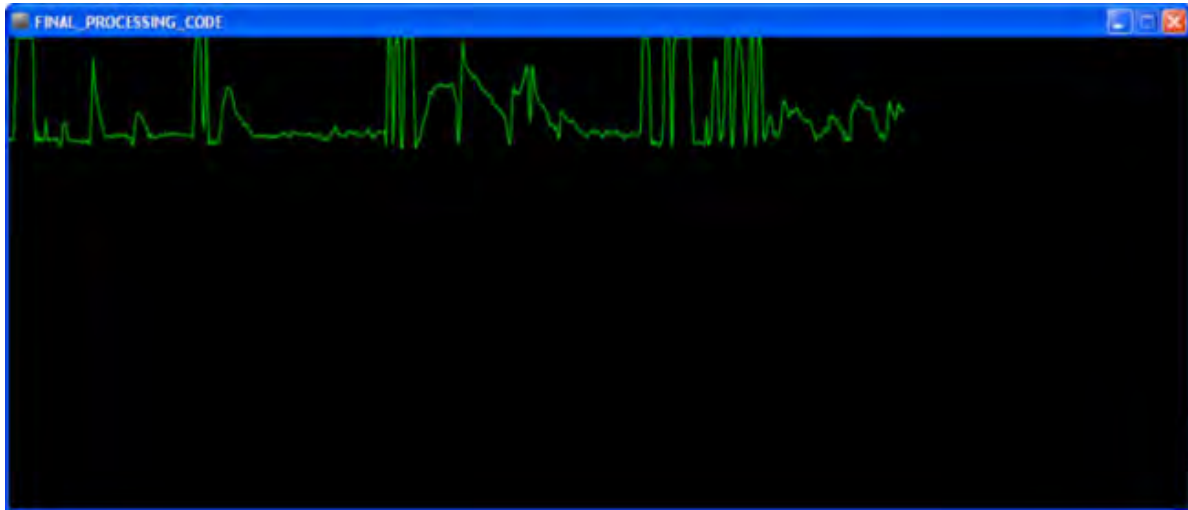
My PC board was pretty messy. After hours of soldering and desoldering and checking diagrams, everything was perfectly in place and the LED lit. I was never more excited to see a light glowing in my life. From all of my mistakes throughout building the board, I learned how to troubleshoot a project.



Step 6: Downloading Software

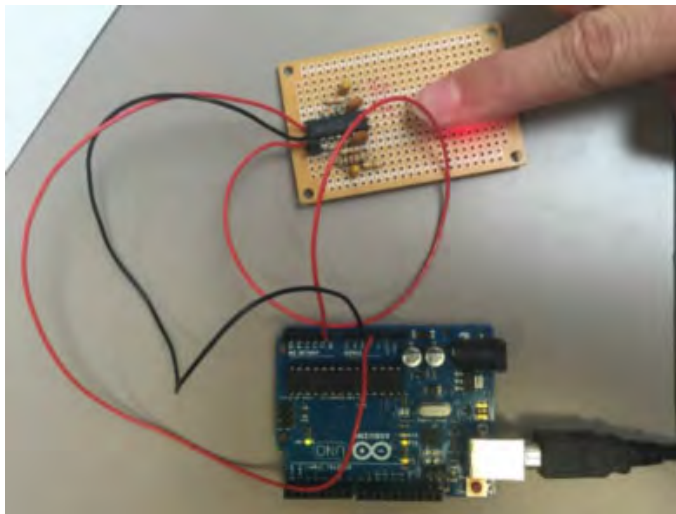
Once the board is ready, download and install the [Arduino software](#). For this project, I used Arduino 1.0.5 but other versions should work as well. In the Arduino software, select the board type (Arduino Uno) and the COM you are using. To make sure my board was successfully connected, I clicked File→Examples→0.1 Basics→Blink. After uploading the code to my Arduino, the LED on the Arduino began to blink, as expected. Download the [DIY ARDUINO PULSE SENSOR](#) sketch and upload the code to your Arduino. Open the serial monitor tool (Ctrl + Shift+M) to see the data coming from the sensor.

After the Arduino sketch successfully uploads to the board, exit the software. In order to visualize your data, download [PROCESSING \(32-BIT\)](#). Then, you can download the [PULSE SENSOR PROCESSING CODE](#). When the code is in Processing, press the play icon. A display should pop up with a black screen and a green line moving across the screen, similarly to an EKG machine.



Step 7: Using the Pulse Sensor

Place your finger over the LED and LDR. Although the board has a few odd readings every now and then, you should get fairly consistent results. Try to minimize the amount of external light touching the LDR so you can get a more accurate reading. After just a few minutes playing around with it, you'll be able to get great readings in a heartbeat!



Please Note: Although this device can monitor your pulse, it is not an approved medical-grade device. For health or life-critical applications, make sure to use an FDA approved device monitoring device.