The original post on the Arduino Forum can be found here:
http://forum.arduino.cc/index.php?topic=256759.0

The goal of this post was to inspire people who may not have any experience with Arduino or DIY at all to take the initiative and make something awesome for themselves.

The Arduino Controlled, Phone Operated, House Wide Audio System

So you want audio huh? But not just any audio, you want audio EVERYWHERE! In your bedroom, bathroom, kitchen, closet, shed, doghouse, etc. Still not enough? Well, what if I told you all of this audio can be controlled (volume and what is playing) on a room by room basis from ANYWHERE in your house, all for far less than it would cost you to buy a similar retail system. Don't believe me? You're not alone, but if you keep reading, hopefully I'll change your mind.

First of all, let me start with some disclaimers: This system is not perfect. There are limitations. Furthermore, it will take effort and it will be worth it.

Second: You will need a building in which you can/already have routed speaker wire to every room in which you want audio. This is ridiculously hard to do in older buildings unless you are willing to tear down walls. I recommend waiting until you plan on renovating/building a house or, I guess, if you're that kind of person, you could just run speaker wire through the hallways and trip over it every time you get up to pet the cat. If you cannot/are not willing to do this, step away now. You will not want to keep reading and realize what you are missing in your house. If you do, I am not responsible for any damage to walls that may occur from spontaneous urges to run speaker wire.

Ok, now that all the boring stuff is over, LET'S GET BUILDING!

PART 1: THE BASICS

The first thing you will need to do is go buy an arduino. If you're unfamiliar with Arduino, it's basically a very small, low powered computer which will be controlling our system. Arduinos range from very small to much bigger and more powerful. The Arduino we'll be using in this tutorial is the Arduino Uno R3. The Uno is a middle range, great for beginners Arduino which is very versatile. You can get one for around $20 from your favorite online store.

Now that you have your Arduino, STOP. Drop everything. Take a weekend and get to know your Arduino. Make its LEDs blink. Power it through a 9v battery. Make it talk to you. Take it for a romantic dinner at your favorite restaurant. Whatever floats your boat. The most important thing is to learn what you are working with! There are plenty of Arduino tutorials available online so I won't bother to make one. If you merely just follow my instructions step-by-step and copy my Arduino code exactly you won't learn anything! Sure, you'll have a working audio system but what good is that? What happens if you want to control your blender from your phone? I know how to do it, but that's because I've taken the

time to learn about Arduino and all the wonderful things it can do.

Now that that's over and you have at least a basic idea of what we're working with it's, time to start thinking about what we want our system to do. First thing's first, we want AUDIO! Well that's easy. All we need is an amp, right? Something like one of these, right? WRONG! What we want is something simple, stripped down, and efficient. We don't need any of that airplay or HDMI mumbo jumbo. We'll integrate that ourselves. What we want is a plain and simple amplifier. That's it. An audio signal in, and a speaker output. I'm using (and highly recommend) this one. It has six channels meaning it will be good for up to three rooms (assuming you take the stereo route which I'm guessing most people will).

Next you will need speakers for each room you want audio. I'm going to let you have some freedom on this task. Really any speakers will do. Go find some that fit your price range and room size. I personally picked up these and have been extremely impressed with their sound quality, especially considering their price.

PART 2: THE ARDUINO

Time for controls! We need to decide how to control our system. I said earlier we'll be using our phones to control it, right? Well how might we interface a phone with an arduino.... There's bluetooth, WiFi, and NFC. Bluetooth and NFC both have annoying range restrictions (NFC especially, HA) so let's choose WiFi. In order to send commands from your phone to the arduino, the arduino will need to be connected to your home network. Luckily, there is an ethernet shield for the Arduino Uno which plugs right into the top of your new best friend. This allows the Arduino to communicate with other devices on your home network and send and receive files from its included microSD card. You probably have a microSD card lying around somewhere. If not, buy one. It doesn't need to be big. The file we are going to put on it is only a couple of kilobytes.

PART 3: THE CIRCUIT

So the Ethernet shield allows our Arduino to communicate with your phone, but how can the Arduino control the audio signal going to the speakers? This problem is solved using the fantastic PT2258 IC. This little guy receives commands over the I2C (pronounced eye-squared-see) bus and has six channels of control to match our six channel amp. Basically what this chip does is take six audio inputs and can control the volume from a range of -78dB to 0dB. The various types of commands it can receive are located in the datasheet and will be discussed more later.

We now need to figure out how to wire this little guy up. If you are a beginner, STOP. This is another learning opportunity. Go buy a cheap soldering iron. You will not need a good one for this project. You should be able to find one for under $20. Now go find an old printer, computer, toaster, etc. and rip it apart. Find some circuit boards and have at it. Soldering is tricky and takes quite a bit of practice to master. Find a tutorial of your liking (trust me, there are plenty out there) and learn.

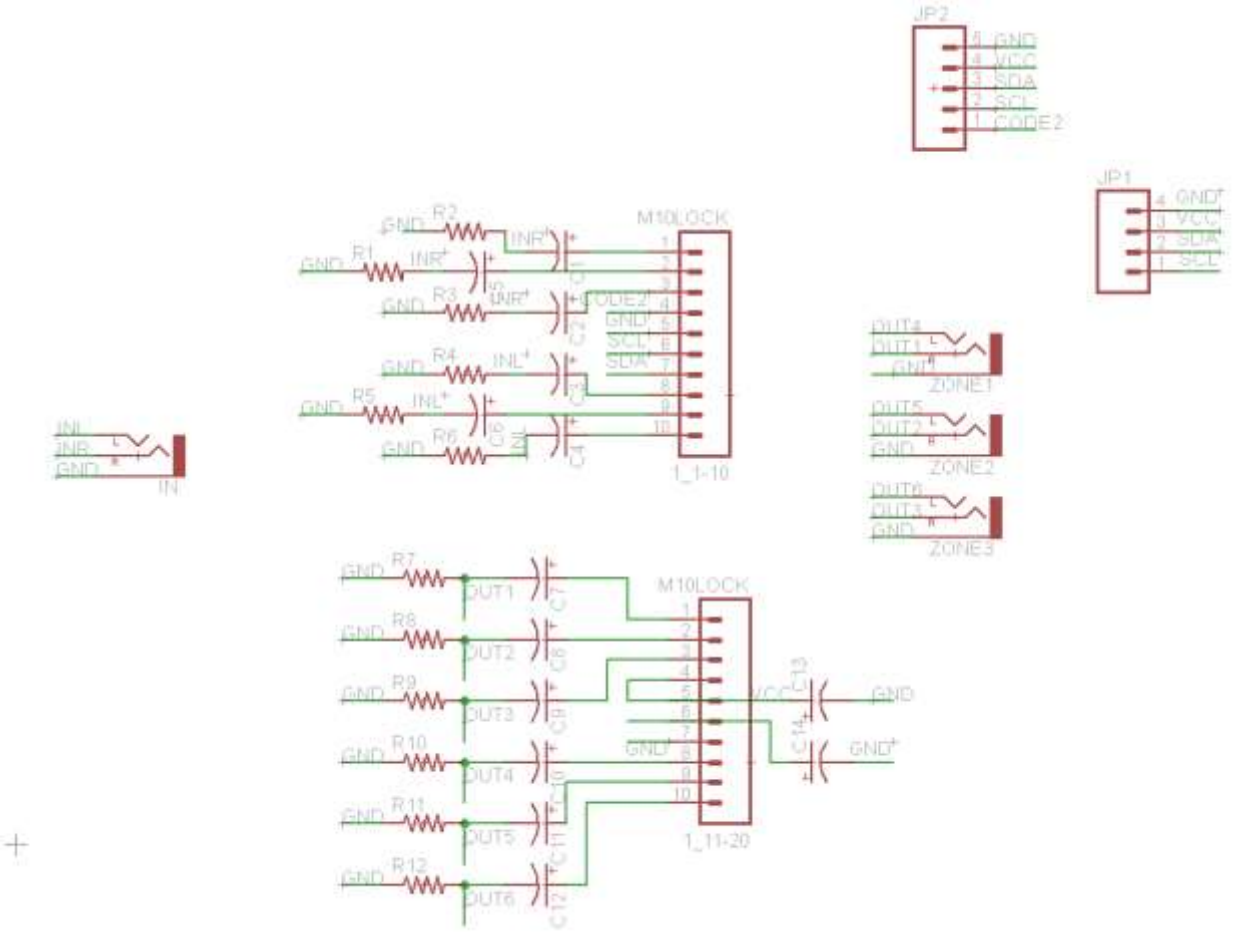
Not too bad, huh? Now you can scratch "Brutally tear apart and defile a toaster" off your bucket list.

Time for a choice. There are two routes you can take for hooking the PT2258 chip up to your Arduino. You could buy a prototyping board such as this and wire it yourself, or you could do as I did and design your own PCB to make things easier and neater. (I have attached the board file for those of you who don't feel like designing your own) Either way you are going to be following the schematic in the PT2258's datasheet EXACTLY. Do not skimp out on capacitors or resistors. These are necessary elements of the circuit and without them you may be severely disappointed in the sound quality.

There are three types of components needed: capacitors, resistors, and the PT2258 IC itself. Each has a different, but equally important role in the circuit. The capacitors store a charge and release it if there is a sudden drop in power from the supply side of the input. When wired in the configuration shown in the datasheet, the capacitors function to smooth the audio signals and remove sharp or sudden spikes. Resistors are generally used to lower voltage or hinder current flow. In the configuration shown in the datasheet, with one end connected to the source and the other to ground, the resistors are called "pull down resistors" meaning when there is no input signal, the circuit is pulled down to ground. This has the handy effect of removing annoying static or buzzing when nothing is being played through the system. If you chose the second method of designing your own PCB, head on over to Sparkfun's excellent Eagle tutorial. Eagle is a piece of software used for designing and manufacturing PCBs.

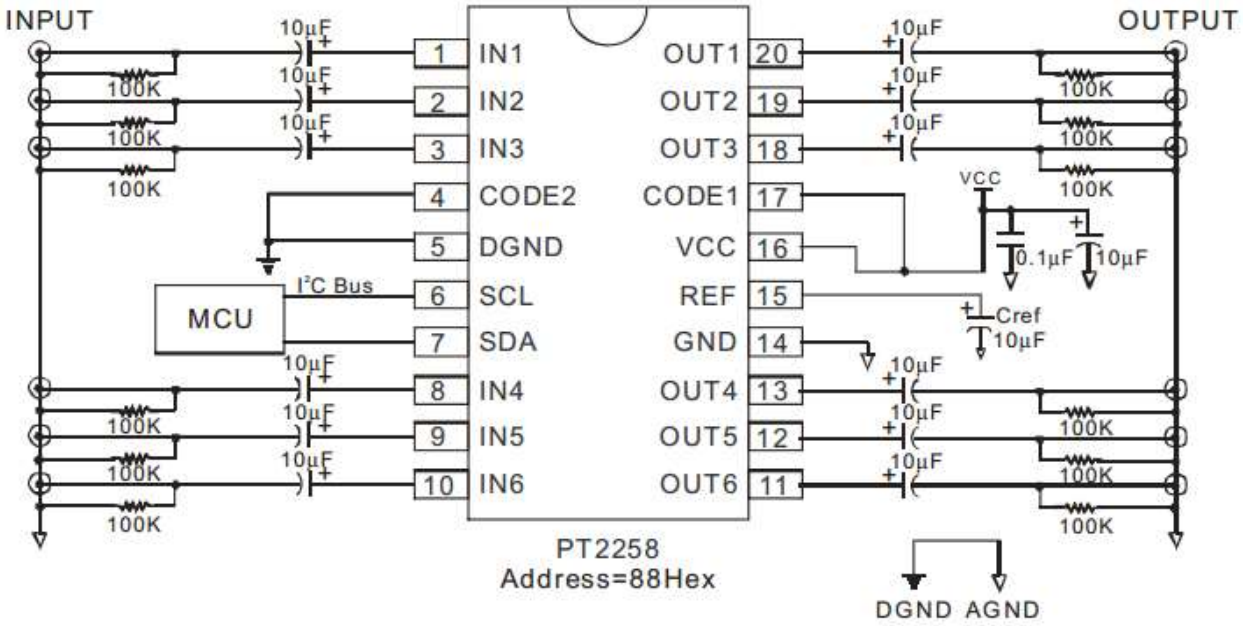
We also need some way to get audio to the PT2258 chip, right? An easy way to do this is to pick up a couple 3.5mm through hole jacks. This allows you to input whatever source into the system you want. We will then need some way to transfer the audio output of the PT2258 chip to our amp. I used more 3.5mm jacks on the PT2258 board and a few RCA to 3.5mm adaptor cables. There also needs to be a way to connect this board to the Arduino so stick a few male headers on the board connected to +5V, GND, SCL, SDA, CODE1, and CODE2 (CODE1 and CODE2 are used to change the I2C address of the chip to allow more than 1 PT2258). I then used these cables to connect the board to the Arduino.

When your schematic is all finished, it should look something like this. The reason I have two sets of male headers is for daisy chaining boards for larger systems.

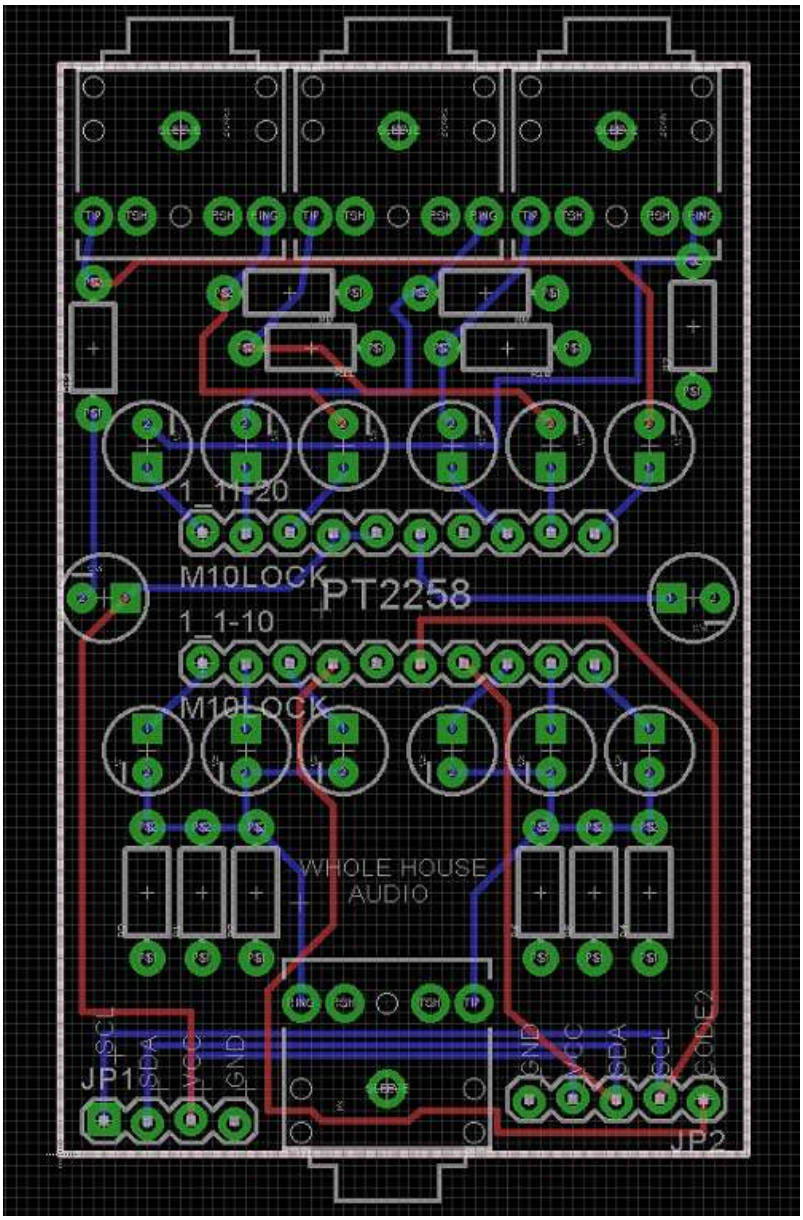


Also notice how the connections in the datasheet match those of the schematic.

APPLICATION CIRCUIT



Next you need to convert the schematic into an actual board. If you used the Sparkfun tutorial you should already know how to do this! Mine ended up looking like this.

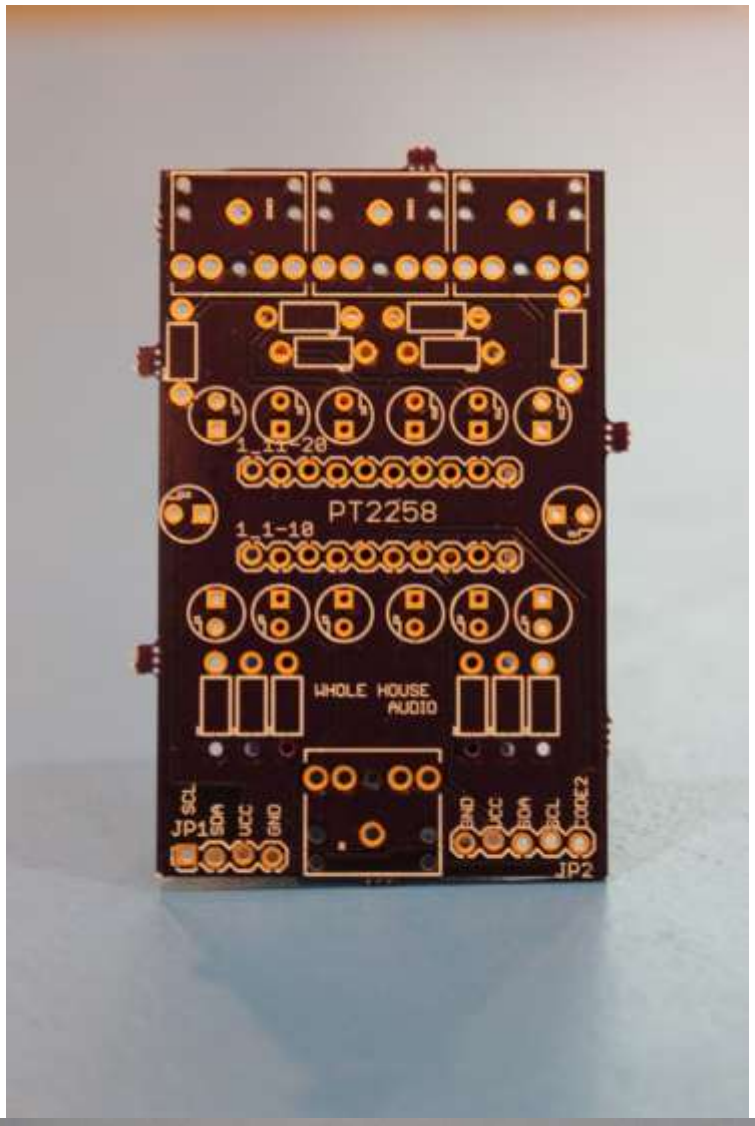


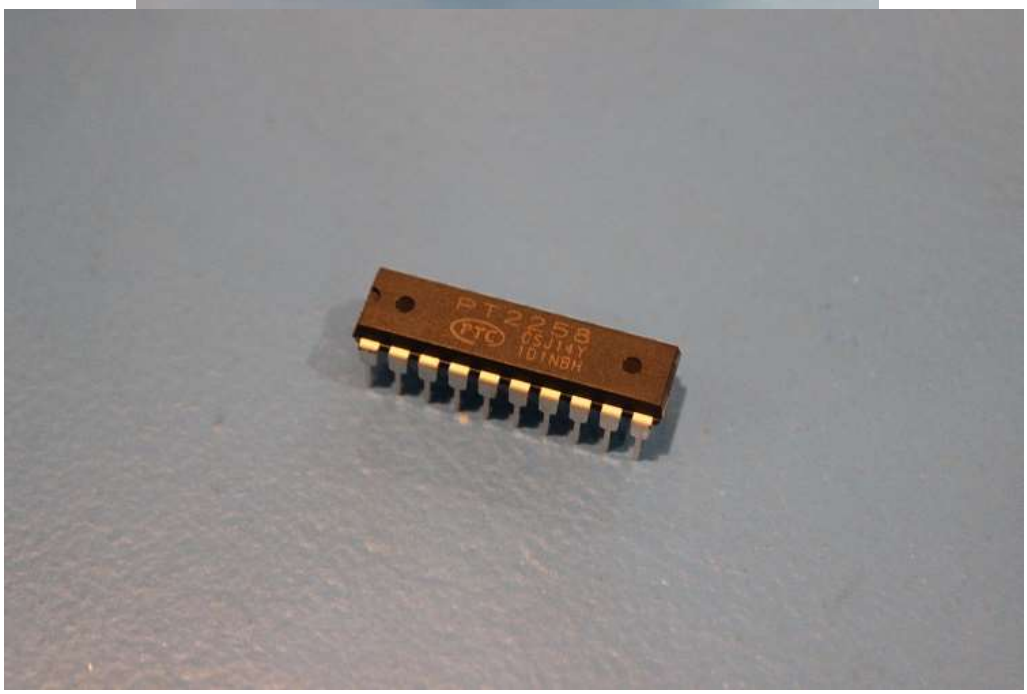
PART 4: ASSEMBLING THE BOARD

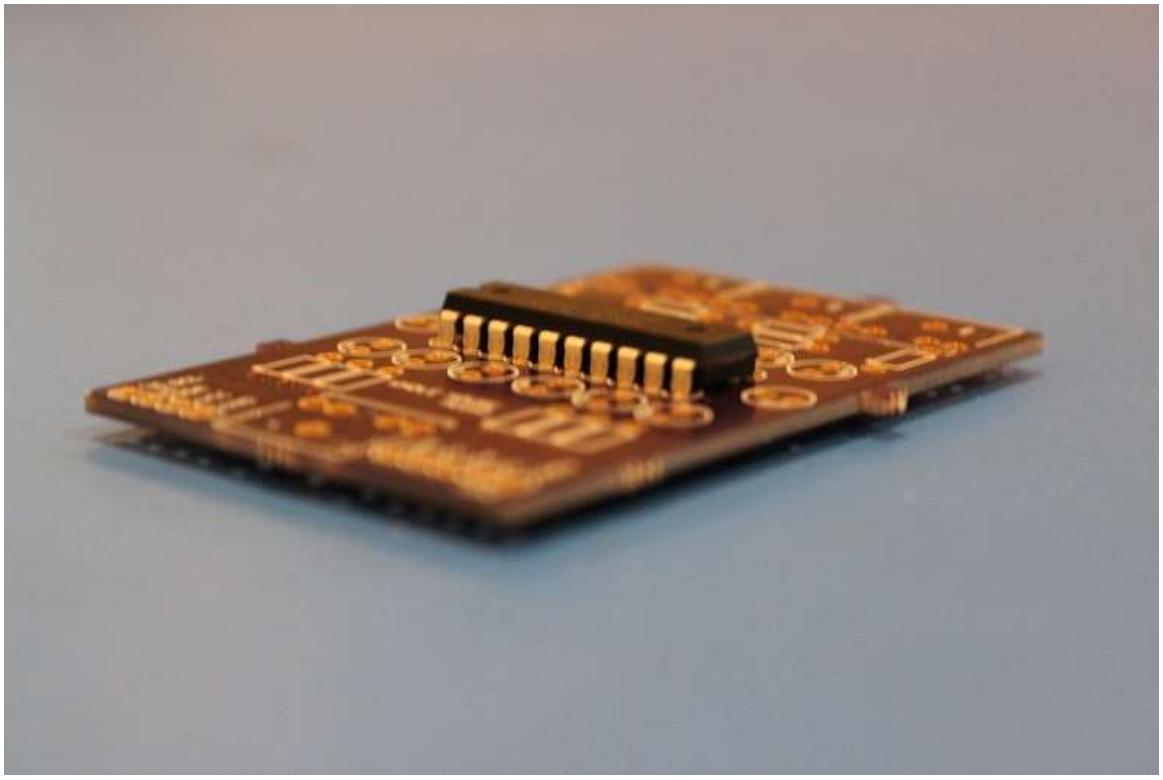
Since we have just finished the schematics for our circuit, we now need to choose our components. The datasheet for the PT2258 suggests 10µF capacitors and 100 kOhm resistors. FOLLOW THE DATASHEET. Order as many of these as you will need for however large your system is. You'll also need to pick up the PT2258 and some male headers. Since you're ordering stuff, if you choose to use the board I designed, head on over to OSH Park and order a couple.

Now for the fun part! It's time to solder! Find that soldering iron and some solder!

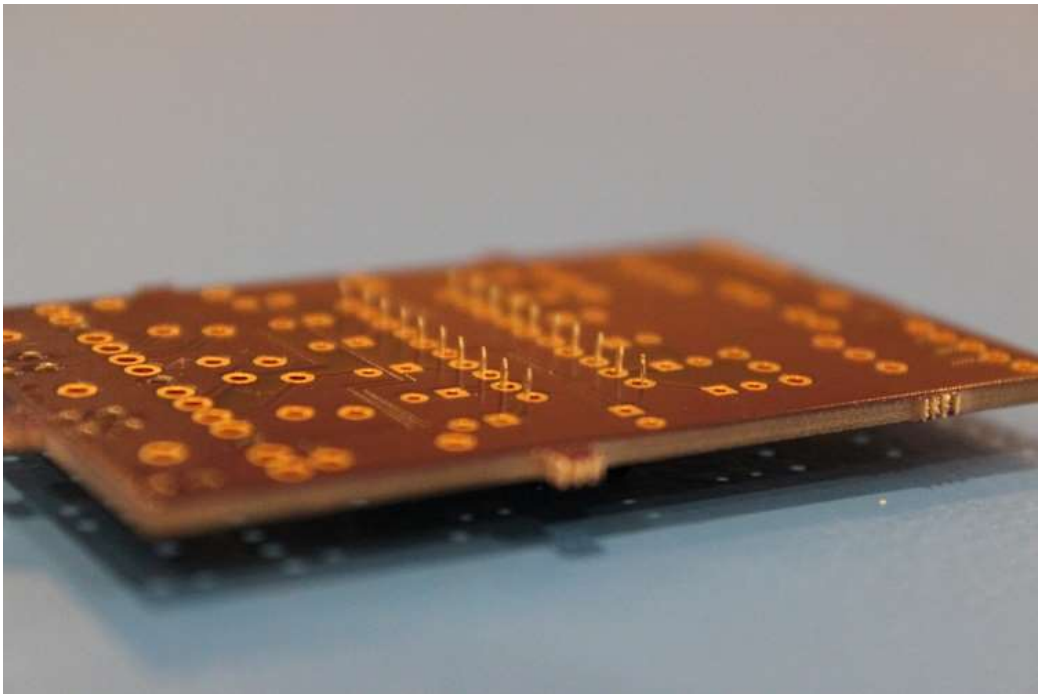
1. Find the PT2258 IC and place it in the PCB



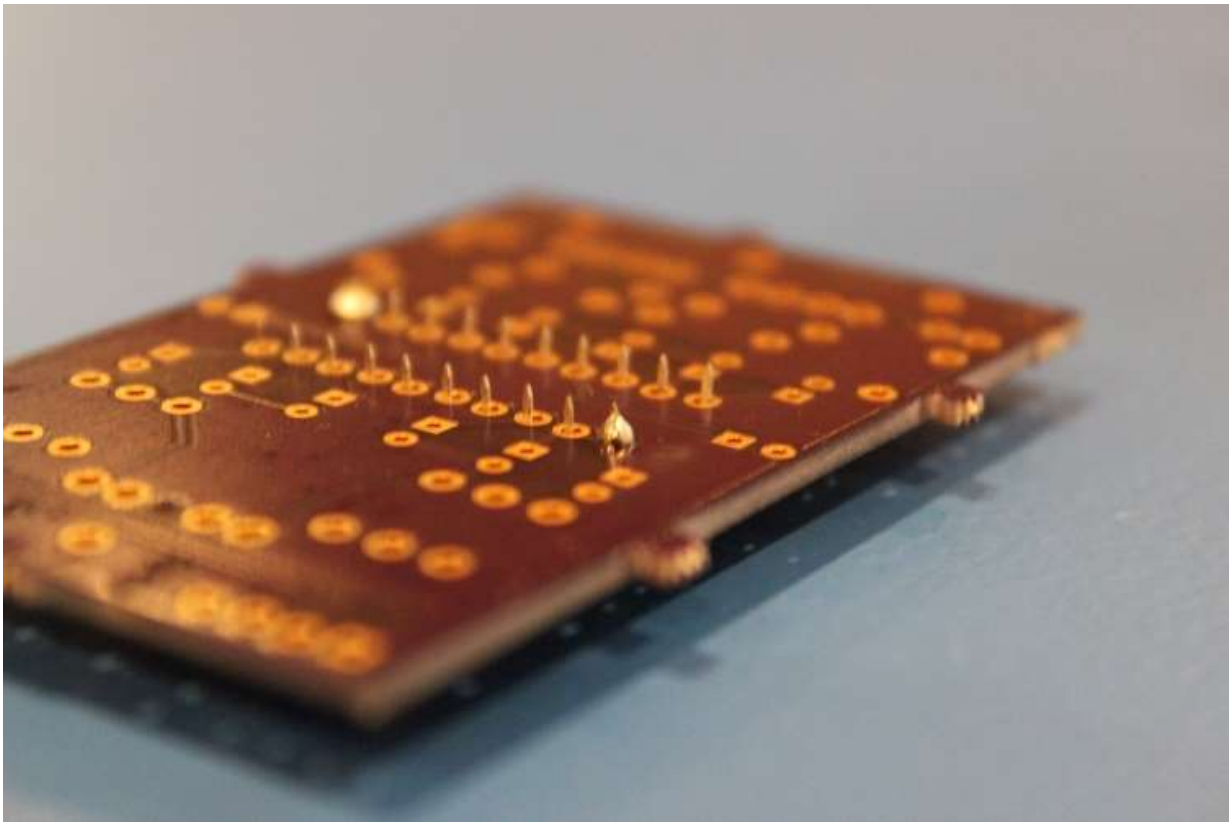




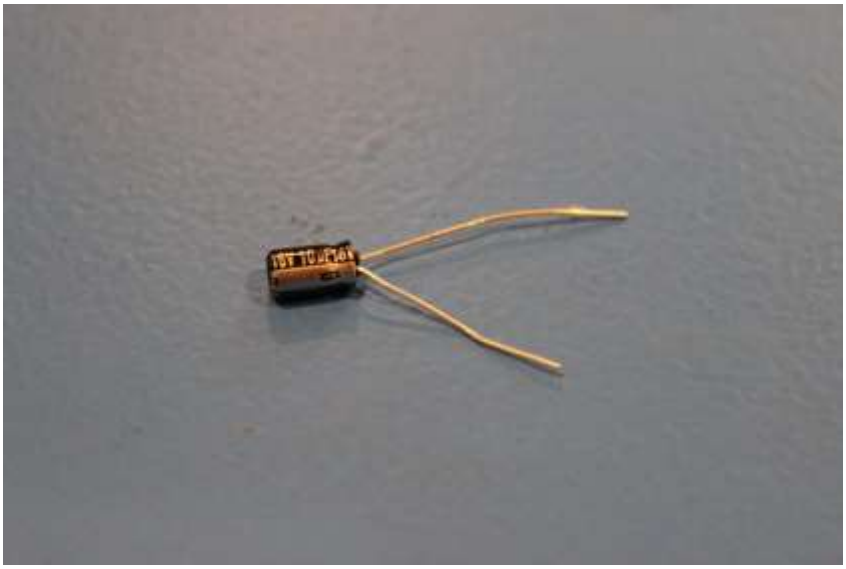
It should look like this from the bottom

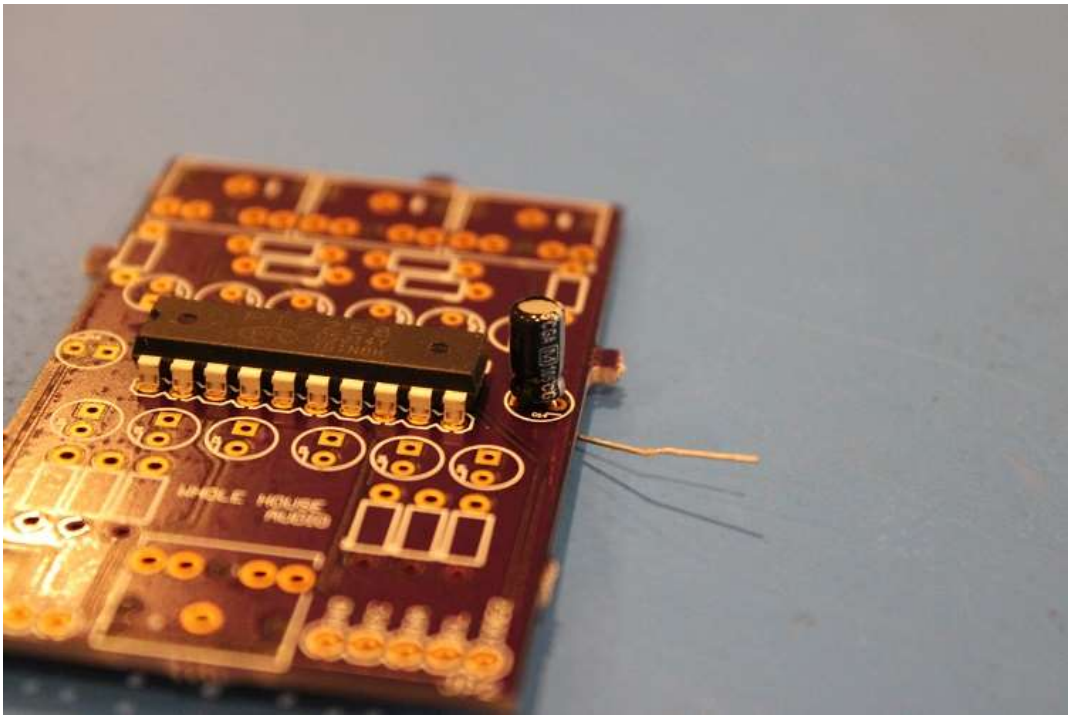


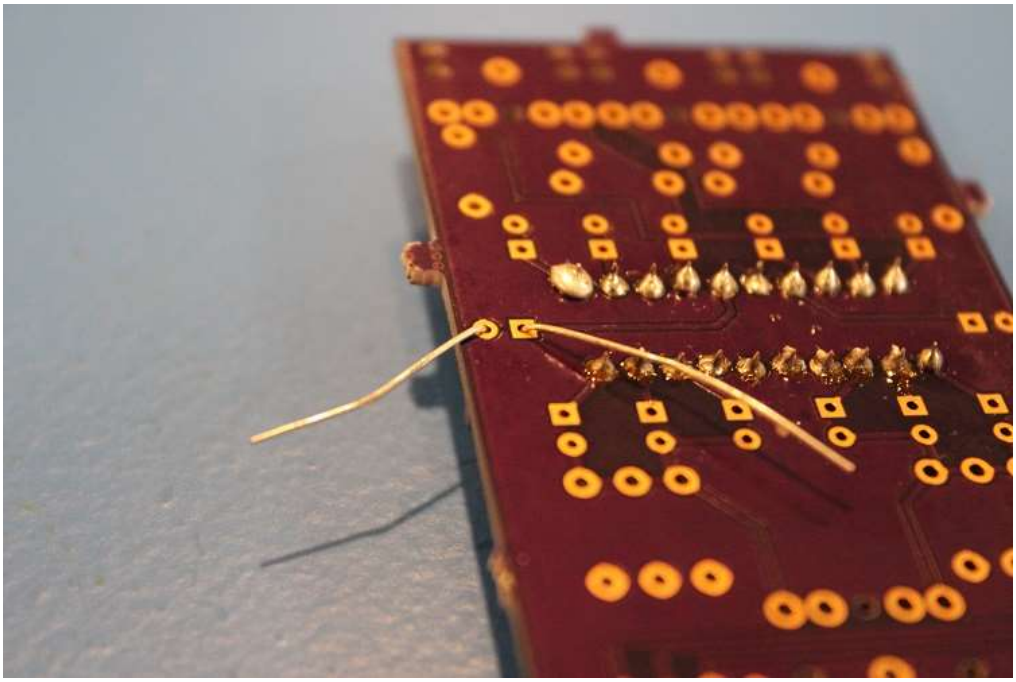
2. Make your first solder joint on one of the legs of the PT2258



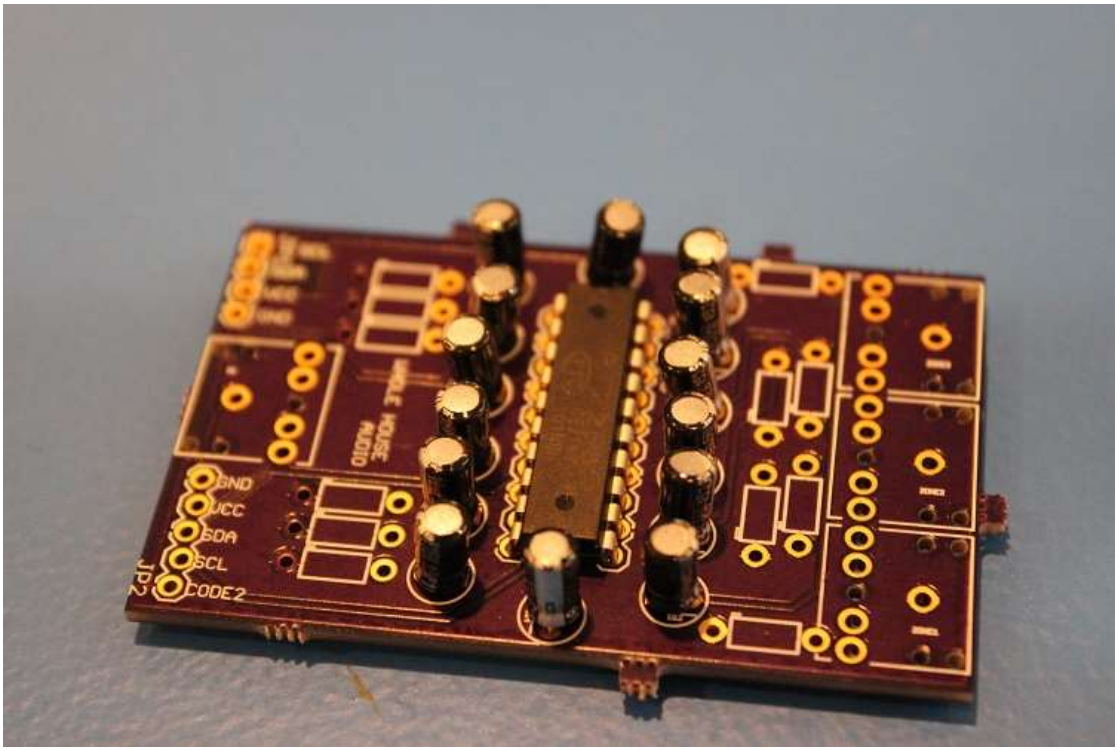
3. Continue making joints until all of the legs are connected
4. Next we will move on to soldering the capacitors. Find the 10uF capacitors and place one in its spot on the board. Bend the leads so it does not move while you solder it.



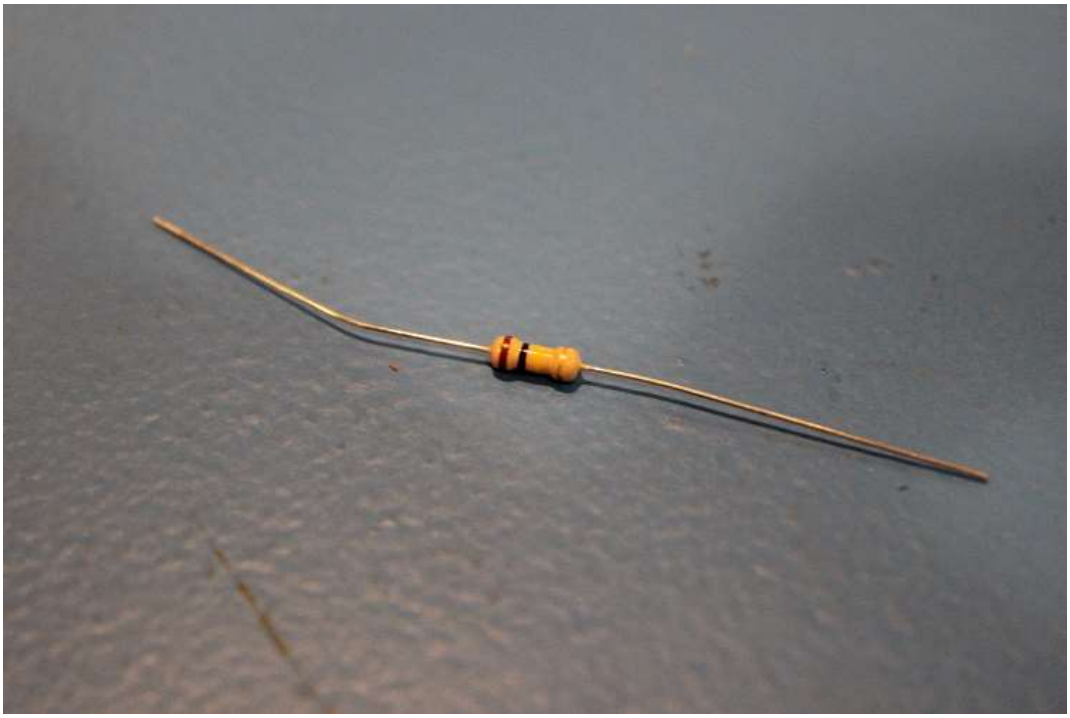


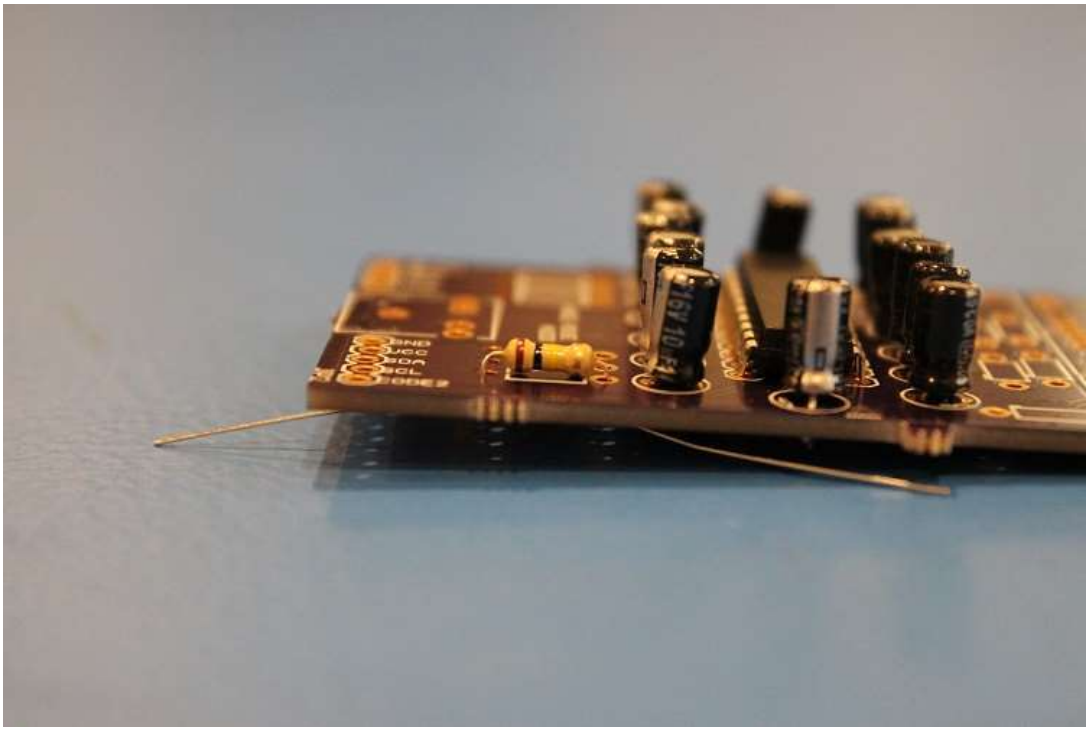


Be careful to not use too much solder on the capacitor joints! You can snip off the excess wire when you're finished.
The board should look like this when all the capacitors have been soldered.

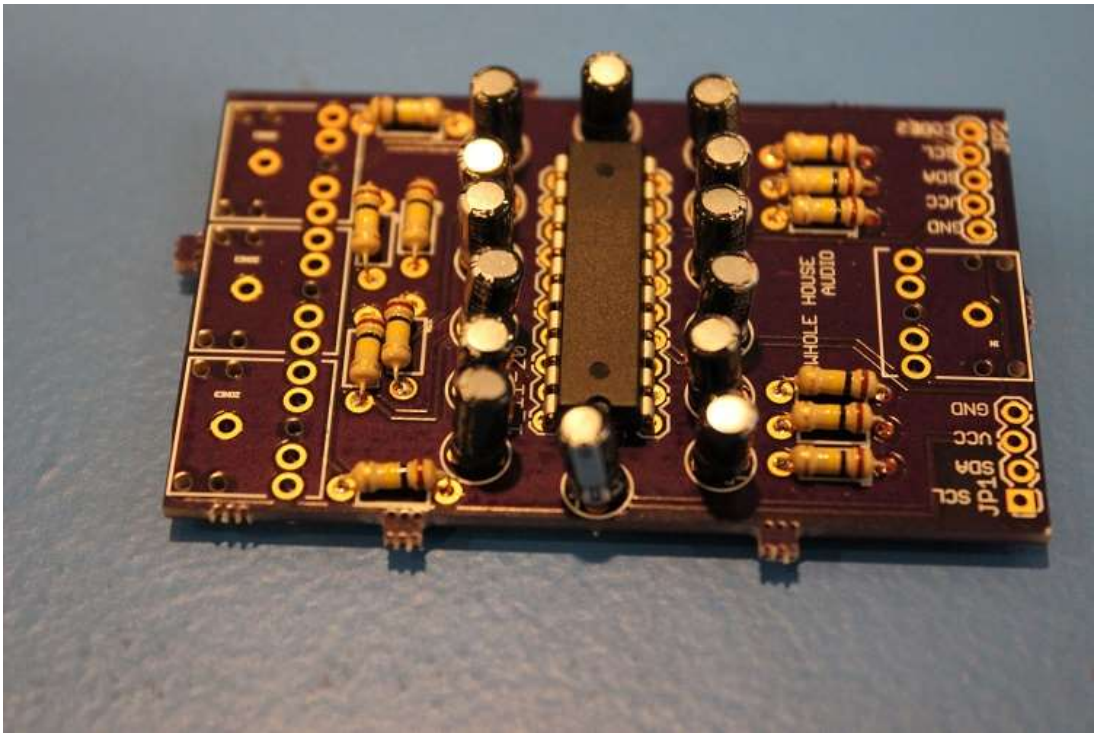


5. Solder the resistors. Put a resistor in its spot and bend the leads as you did with the capacitors.

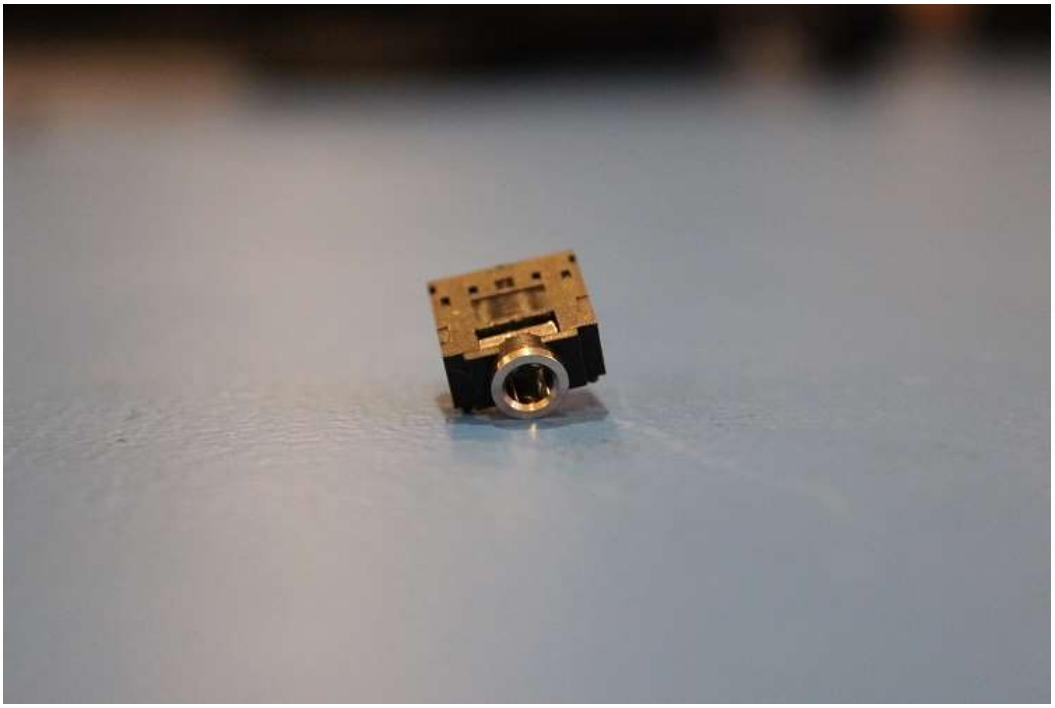


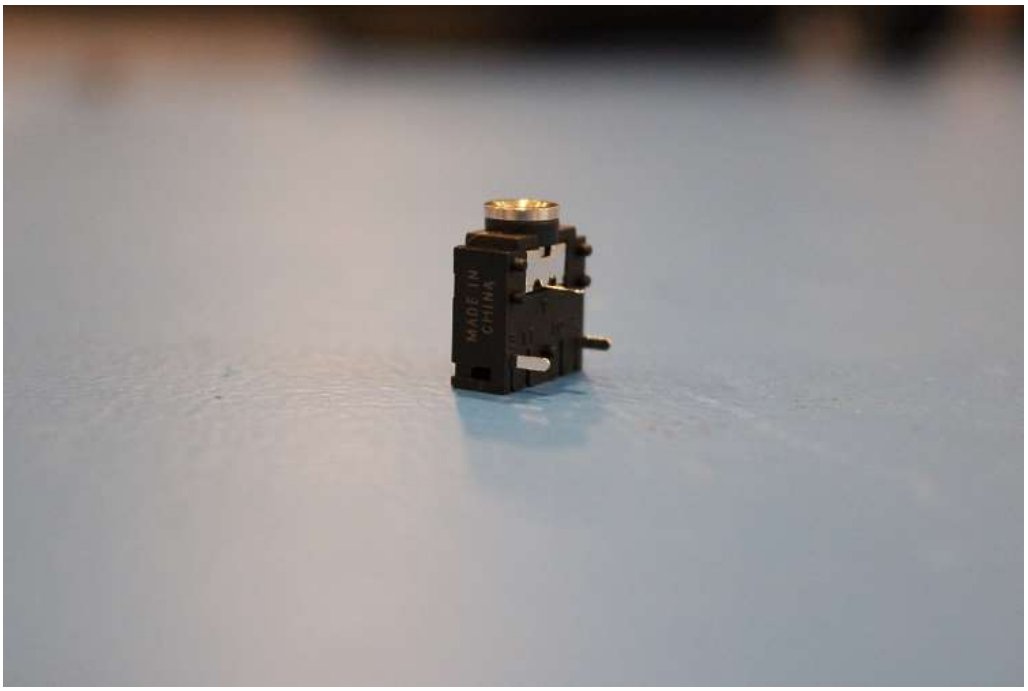


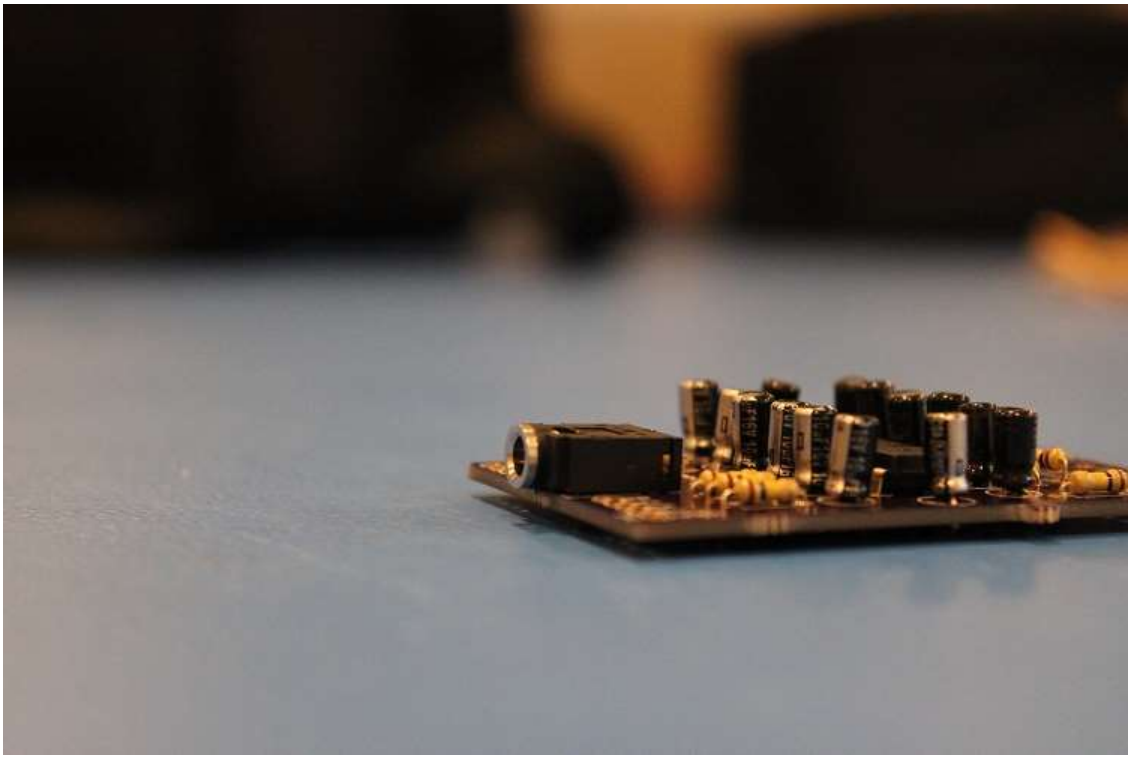
The board will look like this after all the resistors have been soldered



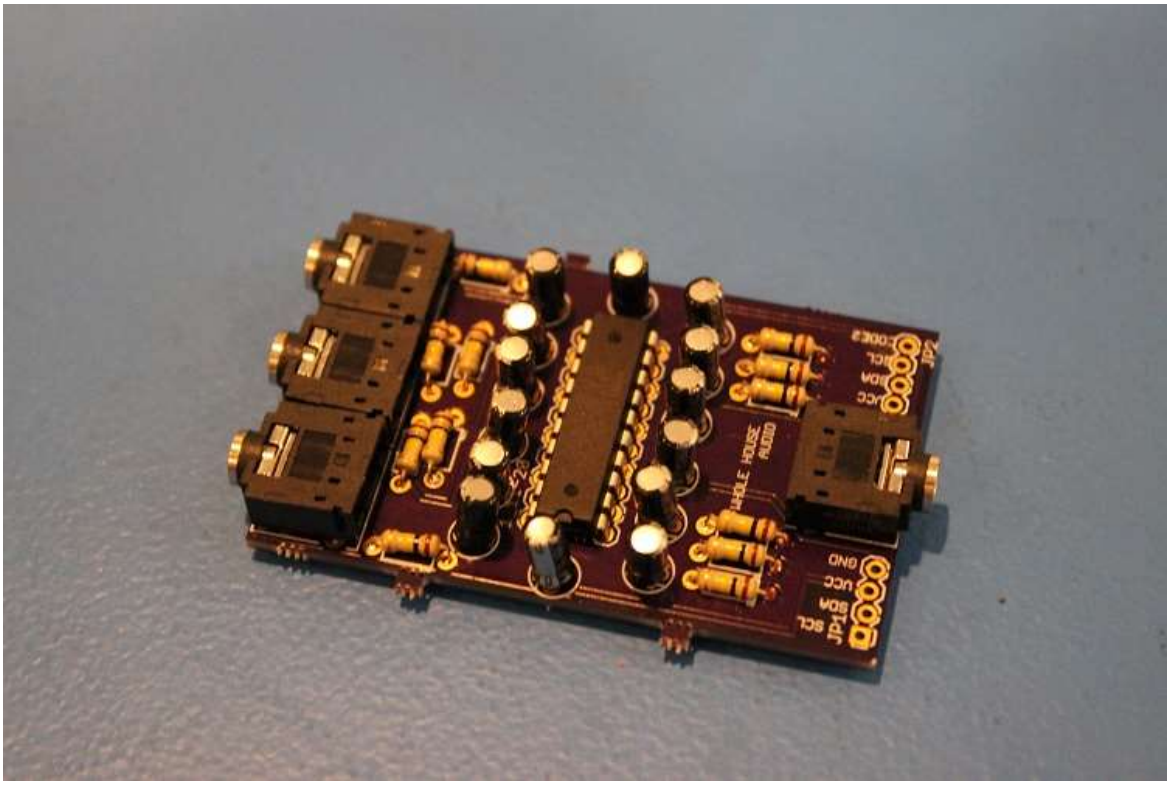
6. Find the 3.5mm jacks and solder them to the board.



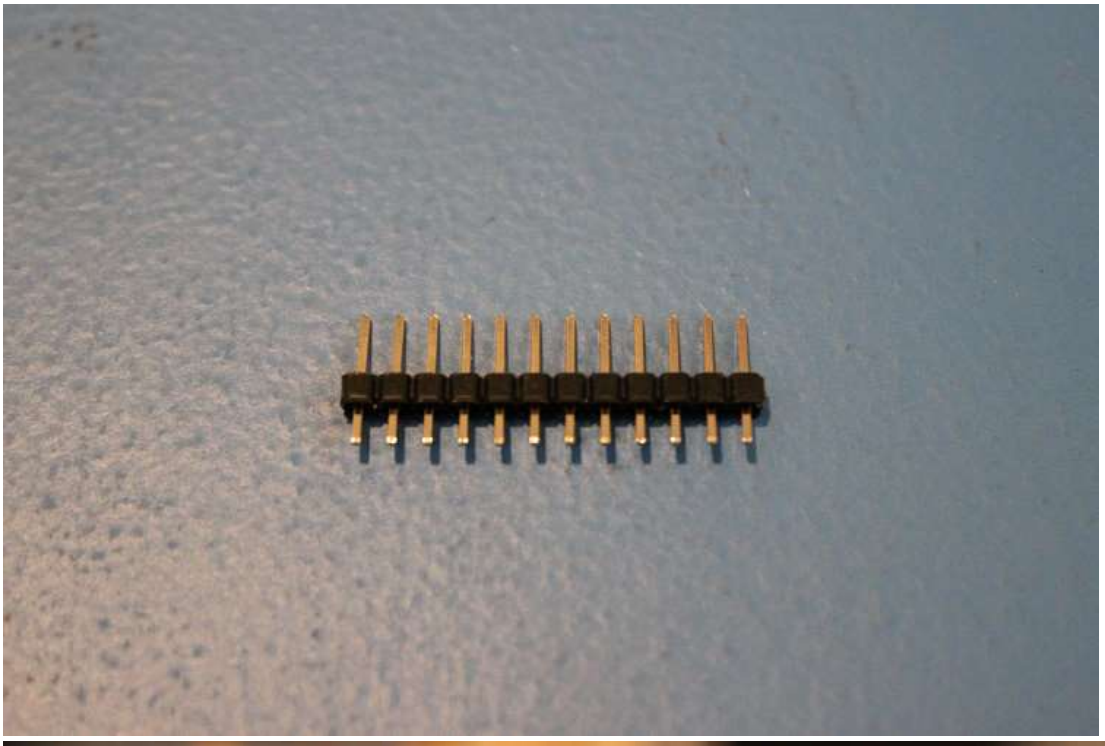

MADE IN
CHINA

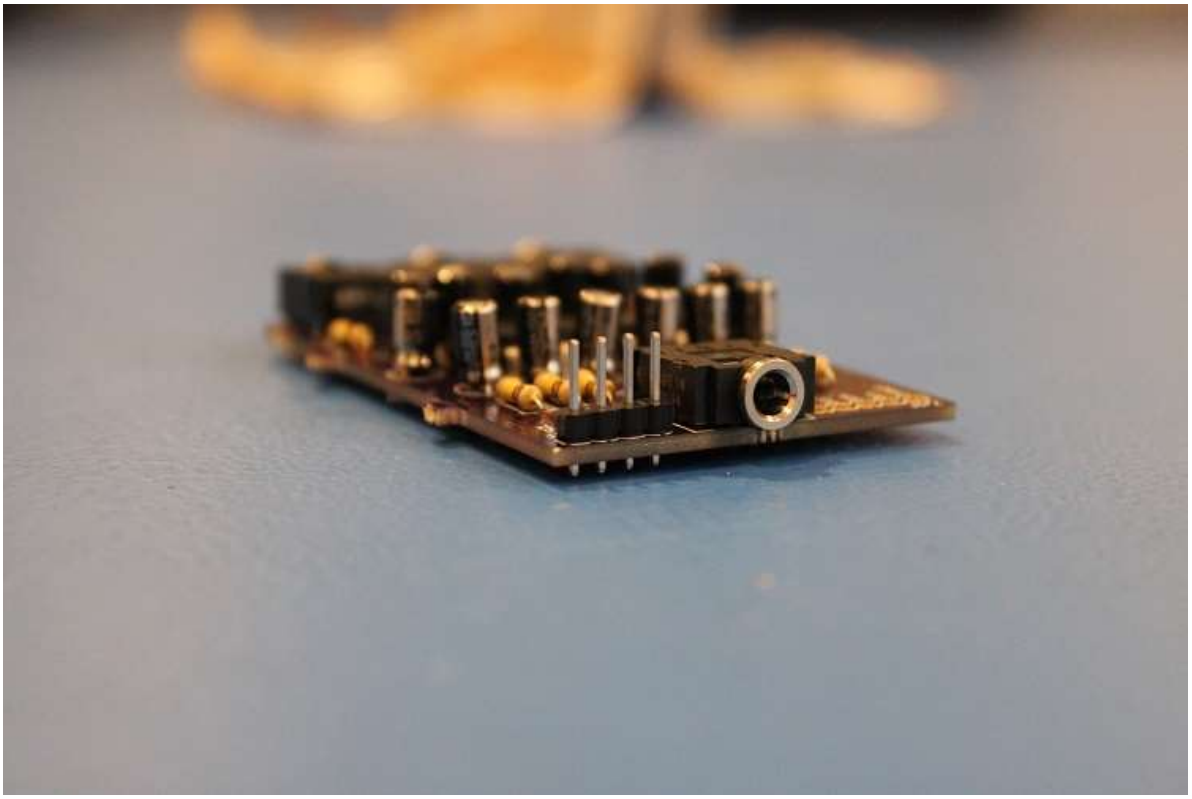


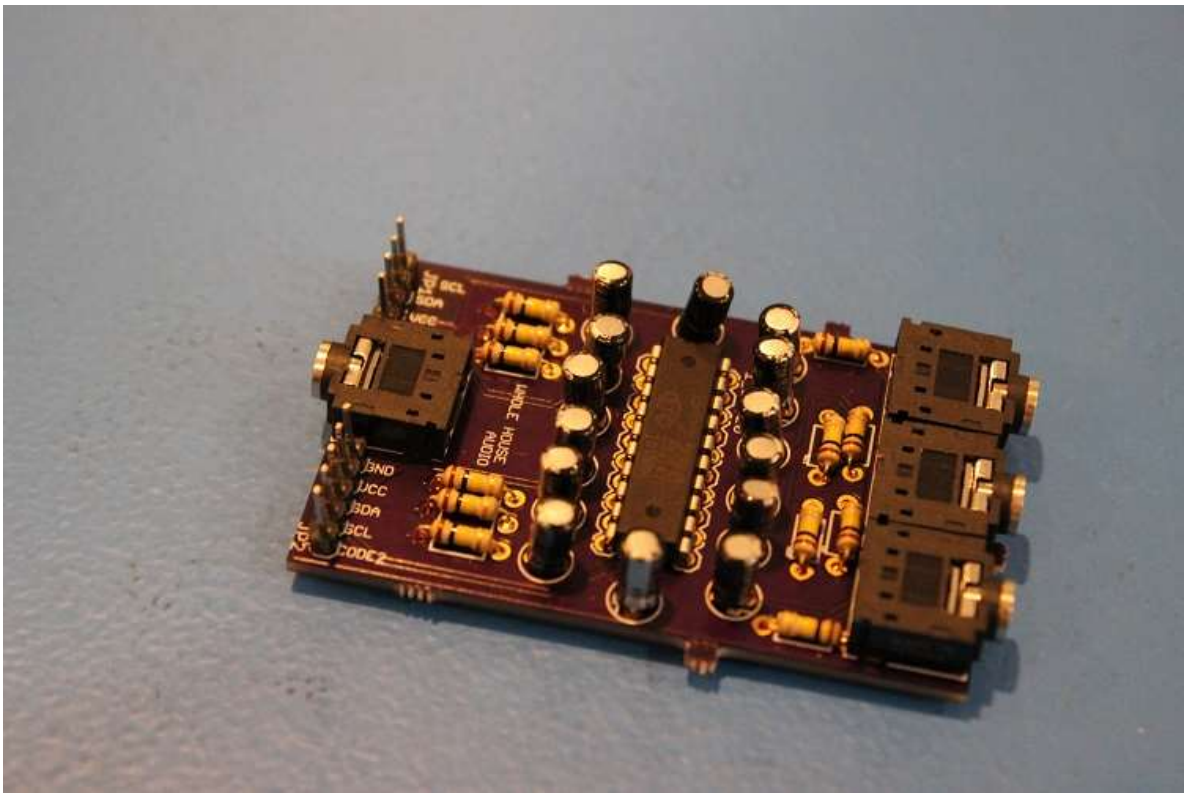
The board should look like this when the jacks are in.

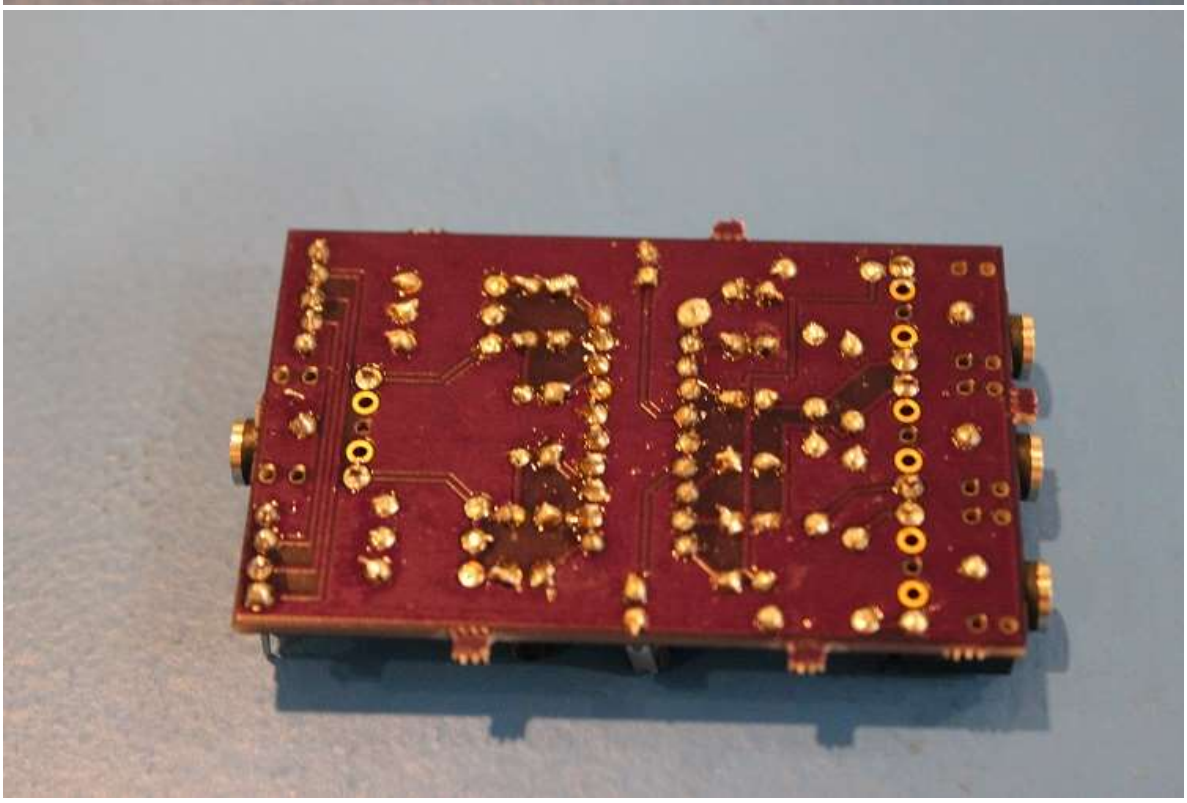


7. Solder the male headers. Break off a length equal to the number of holes in the board and put them in.

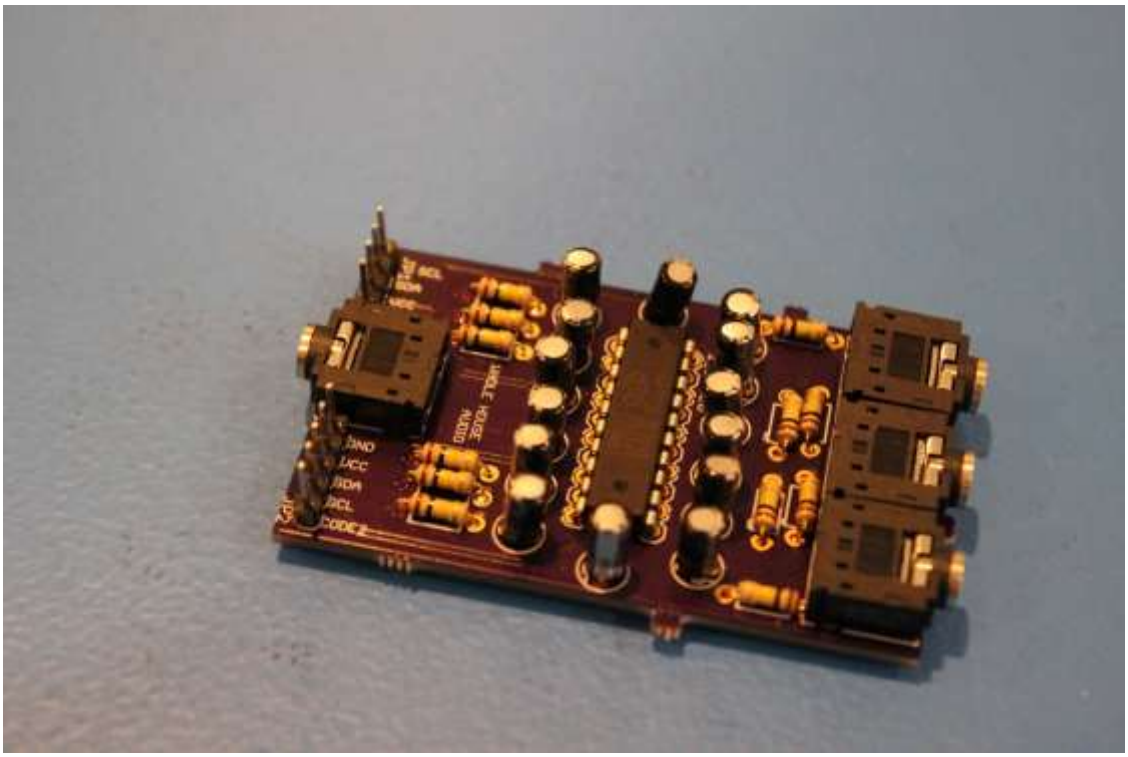








When you're finished it should look something like this.

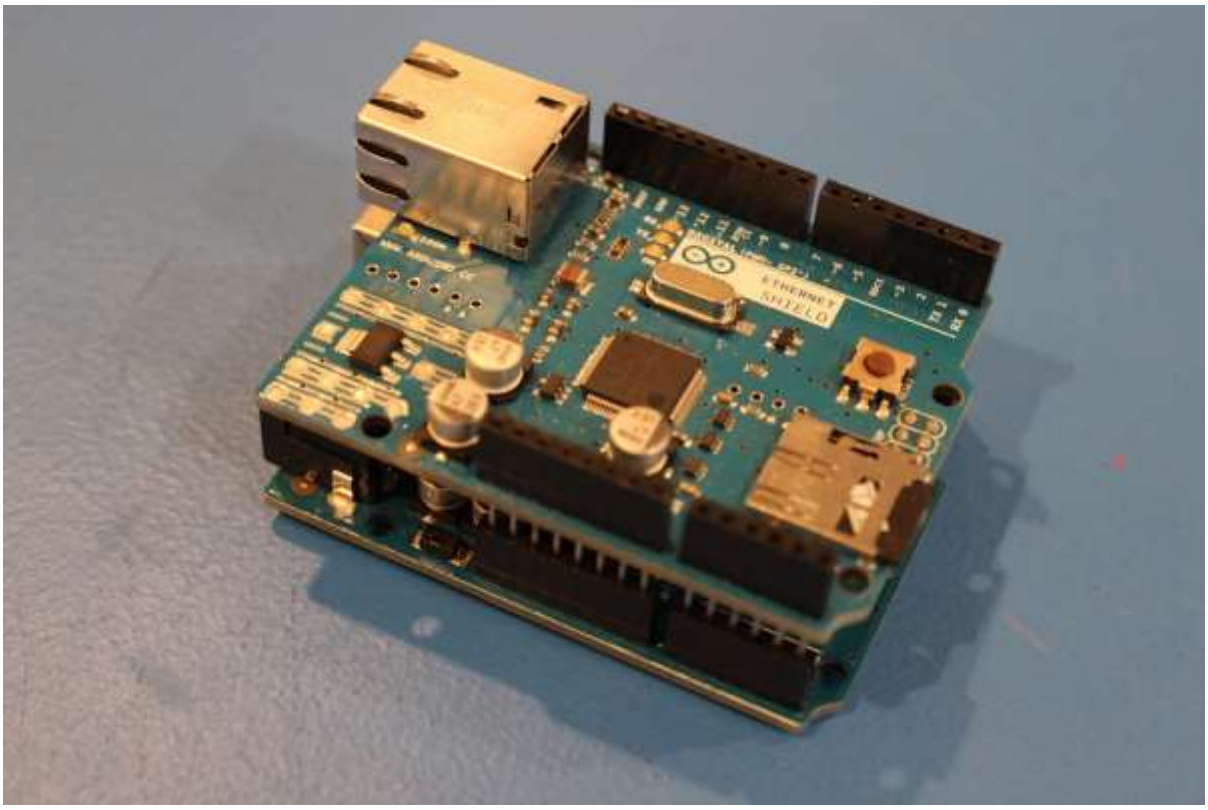


PART 5: WIRING IT ALL TOGETHER

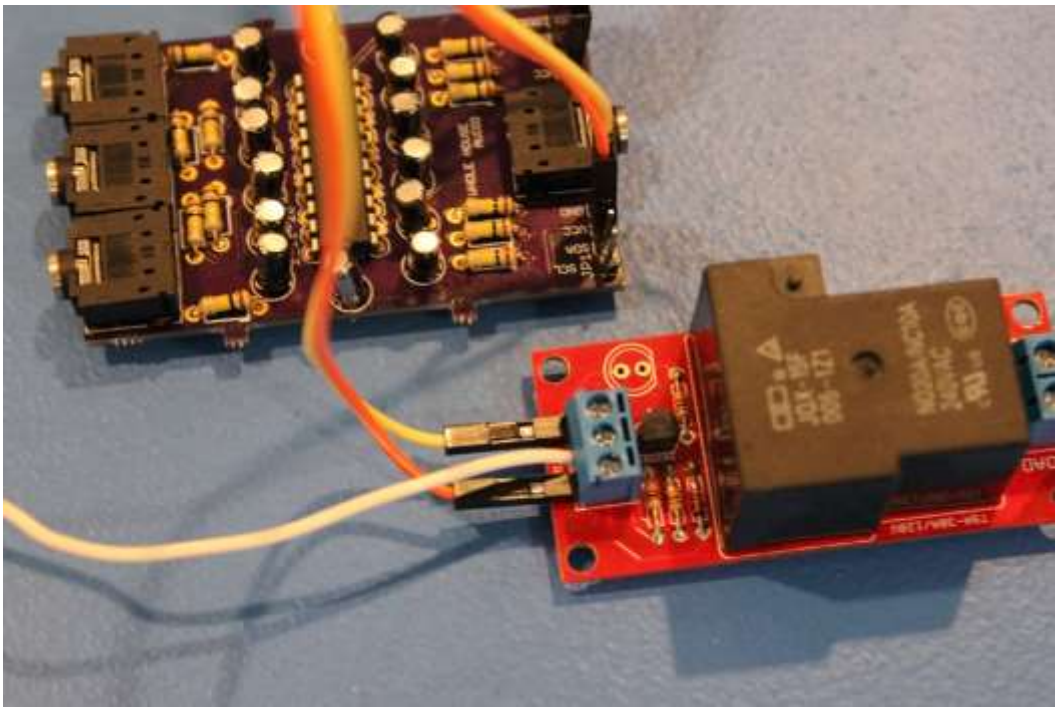
Now that the volume control module is finished, there is one more thing we need to consider. We want to be able to turn the system on and off remotely. I used this relay in line with the power supply for the amp to allow the Arduino to control whether the amp gets power.

Time to wire everything together:

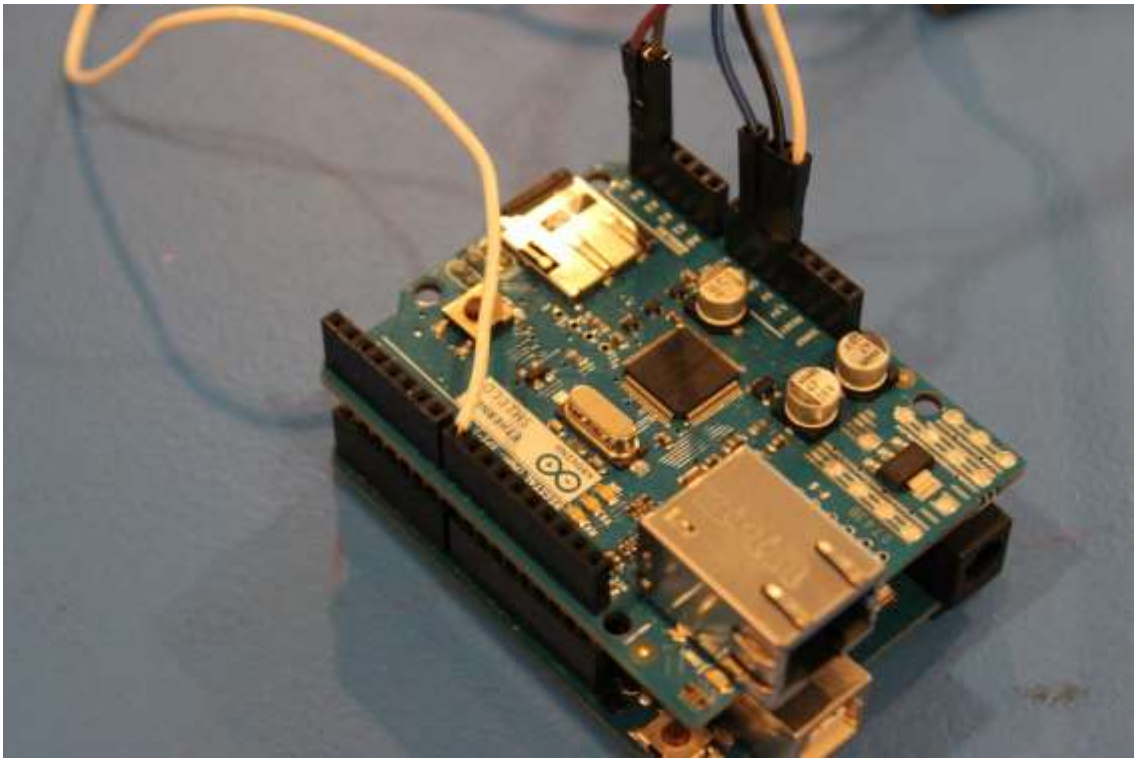
1. Plug the Ethernet shield into the top of the Arduino



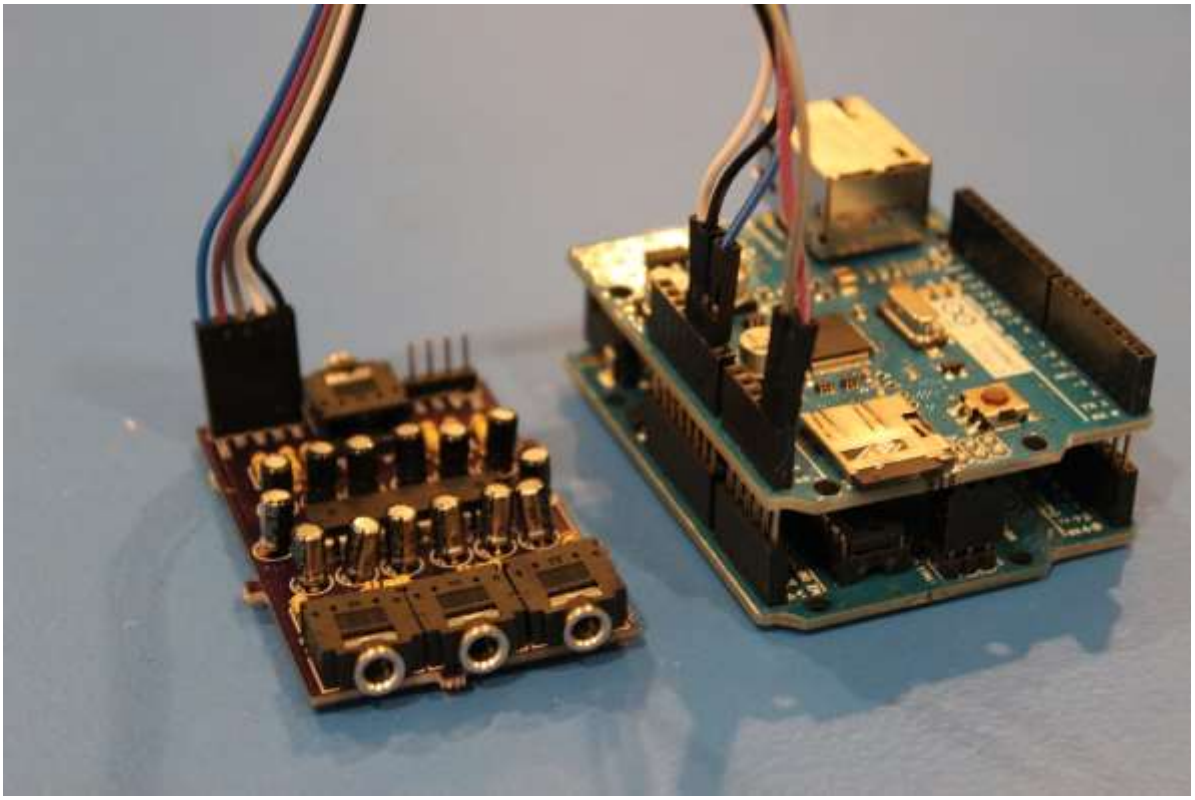
2. Connect 5V and GND on the relay to 5V and GND on the PT2258 board



3. Connect the relay signal to any of the digital pins of the Arduino

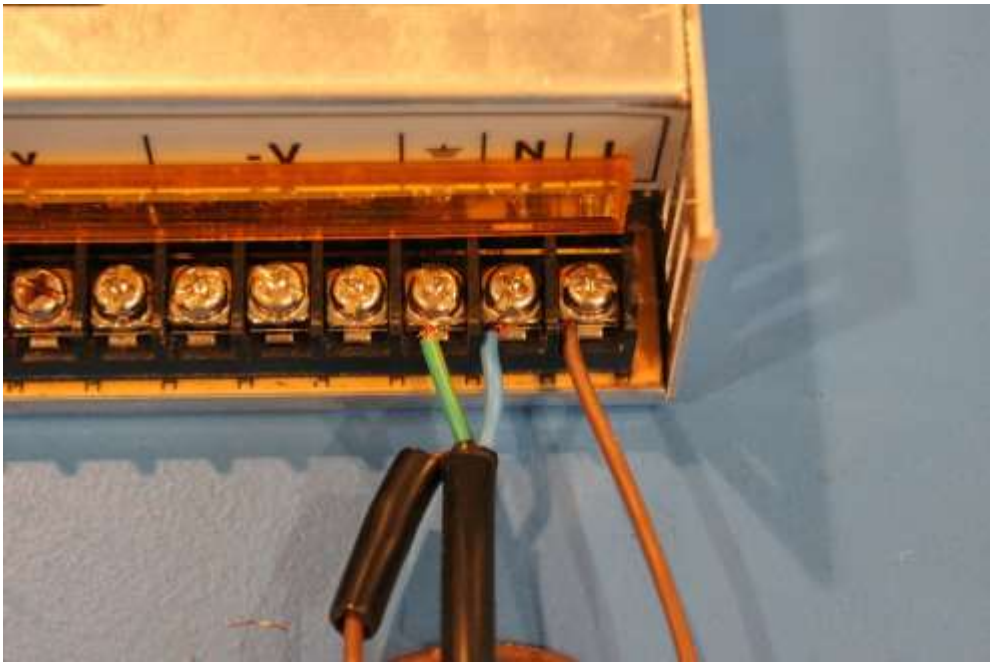


4. Connect 5V and GND of the PT2258 board to the Arduino
5. Connect SCL and SDA of the PT2258 to A5 and A4 on the Arduino, respectively
6. Connect both CODE1 and CODE2 to GND on the arduino (if you are using two PT2258s, connect CODE2 to +5V on the second)

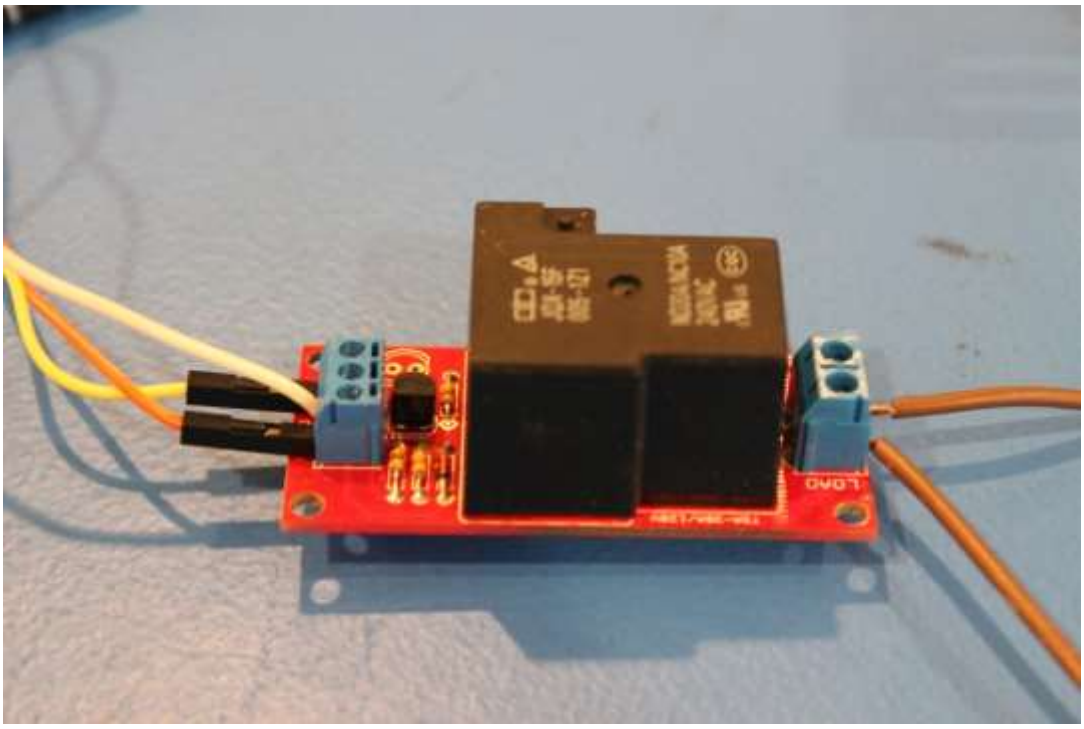


7. Find a spare 3 prong AC power cable (it MUST be 3 prong), cut the non-3-pronged end off and expose the three wires inside.

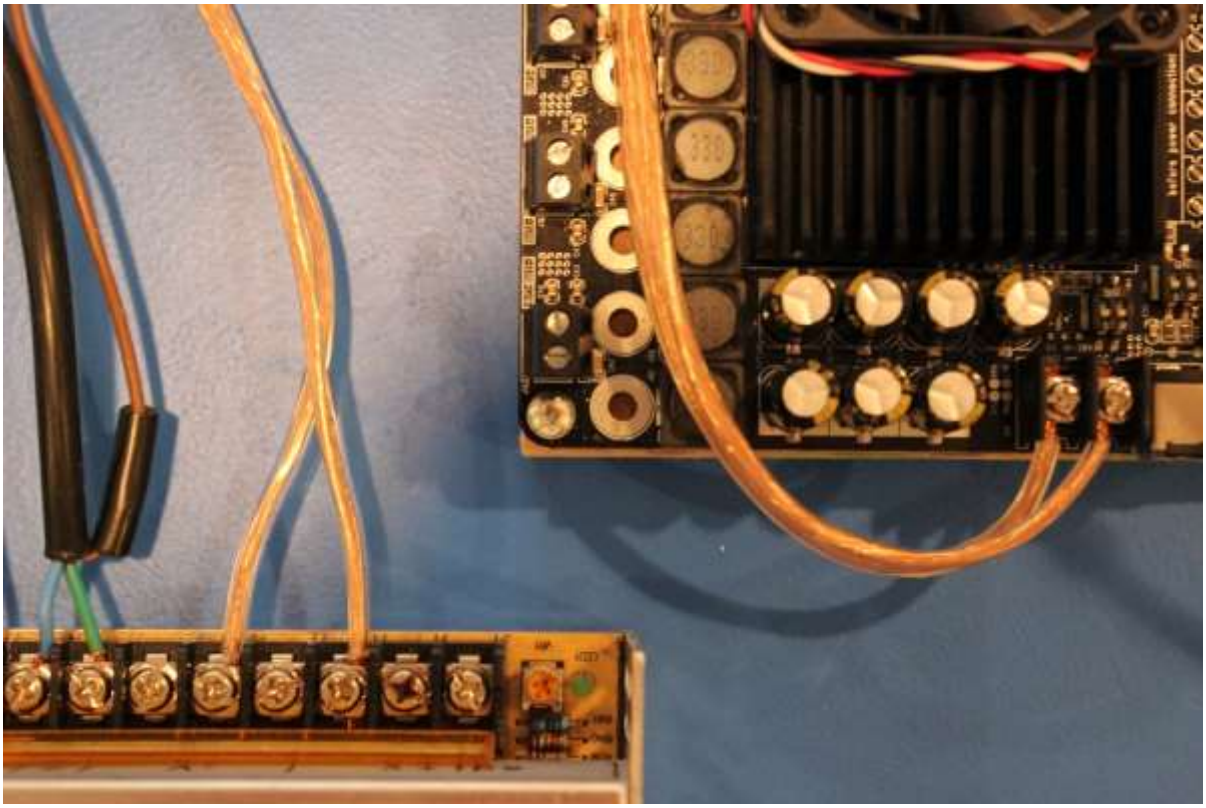
8. Do some Googling to find out which of your three wires is live, which is neutral, and which is ground.
9. Connect neutral and ground to their corresponding ports on the power supply and connect the live wire to the normally open port on the relay board.



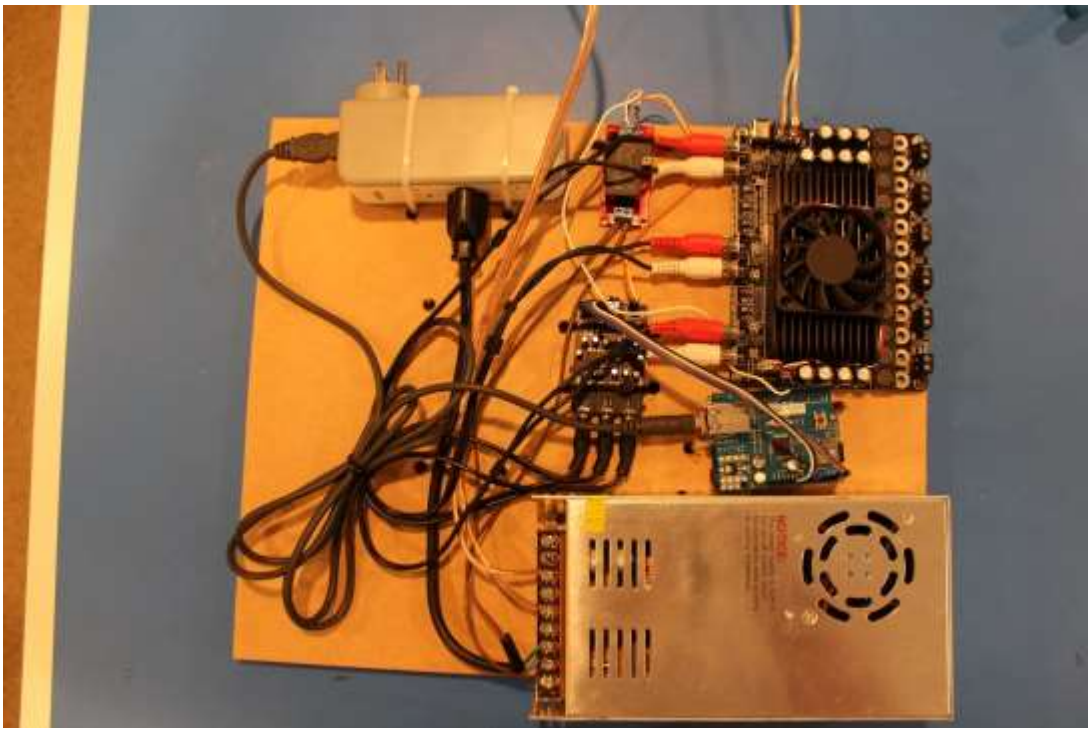
10. Connect the load port on the relay to the live port on the power supply.



11. Using some spare wire, connect + and - of the power supply to + and - on the amp.



The finished system should look like this



PART 6: AUDIO INPUT

What would be the point of all this if you still had to walk over to your computer to change the song? The days of getting out of your La-Z-boy to play "Never gonna give you up" are over!

If you haven't yet heard of a Chromecast, prepare to be amazed. A Chromecast is a small HDMI dongle which allows you to play Netflix, Hulu, Pandora, whatever you want on your TV! All for just $35! Oh, and did I mention you control it from your phone through WiFi? Bingo! That's what we want. If you pair a Chromecast with an HDMI audio extractor, you are able to take the audio signal the Chromecast outputs and send it through our PT2258 chip. All you need to do is plug the Chromecast into the HDMI port on the audio converter and plug the RCA outputs on the converter into the input of our freshly soldered PT2258 board.

Wow, that was pretty easy! Now on to some harder parts….

PART 7: ARDUINO CODE

This section will be relatively short. Writing this code is not a beginner task. If you are well versed in C++, feel free to have a go at writing your own code. For those of you who are still beginners, I have attached the code I have written for this project. Feel free to use it however you like!

There are some changes you must make to the code depending on your setup. I have listed those at

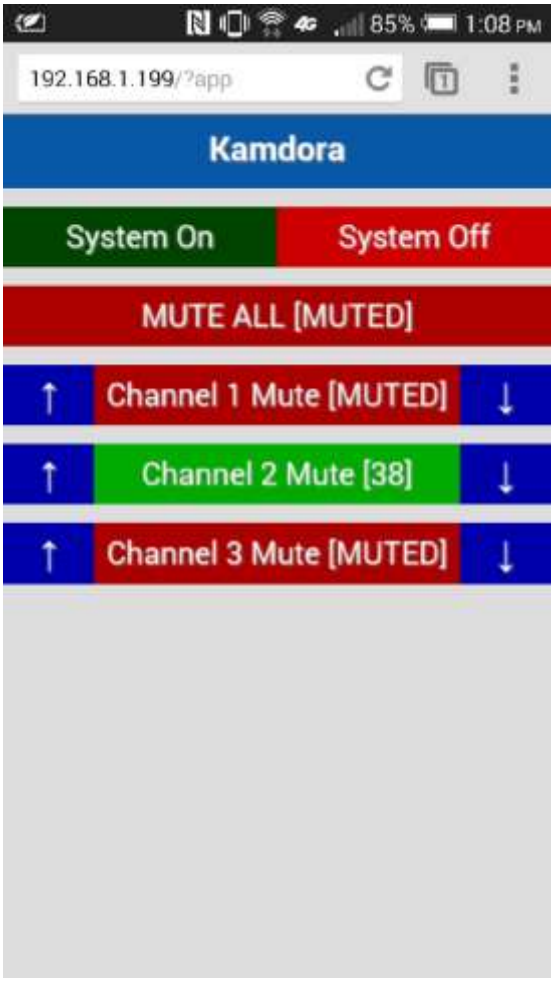
the very top of the code file and have included instructions on how to fill in the blanks.

PART 8: THE WEB-APP

I will be approaching this section in the same way as the Arduino code. I have attached the code I use and, if you are so inclined, you may use it knowing that it will work. All you need to do is take the HTML file containing the webpage, put it on the microSD card in the root directory and name the file "Kamdora". Next put the microSD card back in the Ethernet shield and you're ready to go!

PART 9: USING THE SYSTEM

In order control the system from your phone you will need to open the browser of your choice and type the IP address of the arduino followed by /?app (i.e. 192.169.1.199/?app). This will take you to the control webpage and you are good to go! Start casting some music through the Chromecast and have full control over where and how loud it plays! By the way, this web-app will work on any device will a web browser, not just your phone.



This is the screen you should see. Press System On to engage the relay and allow power to travel to the power supply and amp. Press mute all to instantly mute all channels and silence your house. Click on

any of the channels individually to mute separate zones. The volume scale is 78 (quietest) to 0 (loudest) and is controlled by the arrows.

Congratulations! You have just successfully completed your very own house wide audio system! I hope you enjoyed my tutorial and that you learned something in the process!

The Arduino Code

```
#include <Ethernet.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>

//THINGS YOU NEED TO CHANGE
byte ip[] = { 192, 168, 1, 199 }; //THE IP ADDRESS OF YOUR ARDUINO: CHANGE THIS BASED
ON WHAT IP RANGE YOUR ROUTER USES (MOST PEOPLE WILL BE FINE LEAVING IT AS THIS)
byte gateway[] = { 192, 168, 1, 1 }; //THE IP ADDRESS OF YOUR ROUTER: TO FIND THIS
USING A WINDOWS COMPUTER OPEN A COMMAND PROMPT AND TYPE "ipconfig" PRESS ENTER AND IT
WILL BE LISTED UNDER DEFAULT GATEWAY
byte subnet[] = { 255, 255, 255, 0 };//THIS CAN ALSO BE FOUND BY USING THE METHOD
ABOVE, IT WILL BE LISTED UNDER SUBNET MASK
byte mac[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }; //THIS SHOULD BE ON A STICKER ON
THE BOTTOM OF YOUR ETHERNET SHIELD
#define address 0x44 //THE ADDRESS OF THE FIRST PT2258 IT SHOULD BE THIS IF CODE2 IS
PULLED TO GND
#define address2 0x46 //THE ADDRESS OF THE SECOND PT2258 IT SHOULD BE THIS IF CODE2 IS
PULLED TO +5V
int relayPin = 8; //THE ARDUINO PIN TO WHICH THE RELAY IS CONNECTED
//END THINGS YOU NEED TO CHANGE


Sd2Card card;
SdVolume volume;
SdFile root;
SdFile file;

#define error(s) error_P(PSTR(s))
#define BUFSIZ 100

boolean reading = false;
boolean SystemOn = false;

boolean muted = false;
boolean muted1 = false;
boolean muted2 = false;
boolean muted3 = false;
boolean muted4 = false;
boolean muted5 = false;
boolean muted6 = false;



EthernetServer server = EthernetServer(80);
EthernetClient client;

int volume1 = 78;
int volume2 = 78;
```

```
int volume3 = 78;
int volume4 = 78;
int volume5 = 78;
int volume6 = 78;

void setup()
{
  pinMode(relayPin, OUTPUT);

  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);

  card.init(SPI_FULL_SPEED, 4);

  volume.init(&card);

  root.openRoot(&volume);

  Wire.begin();
  delay(500);

  Wire.beginTransmission(address);
  Wire.write(0b11000000);
  Wire.endTransmission();
  Wire.beginTransmission(address2);
  Wire.write(0b11000000);
  Wire.endTransmission();
  setVolume1();
  setVolume2();
  setVolume3();
  setVolume4();
  setVolume5();
  setVolume6();

  Ethernet.begin(mac, ip, gateway, subnet);

  server.begin();
}

void loop()
{
  checkForClient();
}

void checkForClient()
{
  client = server.available();

  if (client)
  {
    boolean currentLineIsBlank = true;
    boolean sentHeader = false;
    String message;

    while (client.connected())
    {
      if (client.available())
      {
        if(!sentHeader)
        {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();
```

```
          sentHeader = true;
        }

        char c = client.read();

        if(reading && c == ' ') reading = false;
        if(c == '?') reading = true;

        if(reading)
        {
          message = message + c;
        }

        if (c == '\n' && currentLineIsBlank)  break;

        if (c == '\n')
        {
          currentLineIsBlank = true;
          break;
        }
        else if (c != '\r')
        {
          currentLineIsBlank = false;
        }

      }
    }
    delay(1);
    interpretMessage(message);
    client.stop();
  }
}

void interpretMessage(String s)
{
  if(s.equalsIgnoreCase("?app"))
  {
          file.open(&root, "KAMDORA.htm", O_READ);
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();

          int16_t c;
          while ((c = file.read()) > 0) {
              client.print((char)c);
          }
          file.close();
  }
  if(s.substring(1,2).equals("m"))
  {
    String c = s.substring(2,3);
    mute(c);
  }
  if(s.substring(1,2).equals("v"))
  {
    String channel = s.substring(2,3);
    String volume = s.substring(3,5);
    int c = channel.toInt();
    int v = volume.toInt();
    switch(c)
    {
      case 0:
        volume1 = v;
```

```
      volume2 = v;
      volume3 = v;
      volume4 = v;
      volume5 = v;
      volume6 = v;
      setVolume1();
      setVolume2();
      setVolume3();
      setVolume4();
      setVolume5();
      setVolume6();
      break;
    case 1:
      volume1 = v;
      setVolume1();
      break;
    case 2:
       volume2 = v;
       setVolume2();
       break;
    case 3:
       volume3 = v;
       setVolume3();
       break;
    case 4:
       volume4 = v;
       setVolume4();
       break;
    case 5:
       volume5 = v;
       setVolume5();
       break;
    case 6:
       volume6 = v;
       setVolume6();
       break;
  }
}
if(s.equals("?on"))
{
  digitalWrite(relayPin, HIGH);
  SystemOn = true;
}
if(s.equals("?off"))
{
  digitalWrite(relayPin, LOW);
  SystemOn = false;
}

if(s.equals("?getVars"))
{
  client.print("Ok*");
  client.print(muted);
  client.print("*");
  client.print(muted1);
  client.print("*");
  client.print(volume1);
  client.print("*");
  client.print(muted2);
  client.print("*");
  client.print(volume2);
  client.print("*");
  client.print(muted3);
```

```
    client.print("*");
    client.print(volume3);
    client.print("*");
    client.print(muted4);
    client.print("*");
    client.print(volume4);
    client.print("*");
    client.print(muted5);
    client.print("*");
    client.print(volume5);
    client.print("*");
    client.print(muted6);
    client.print("*");
    client.print(volume6);
    client.print("*");
    client.println(SystemOn);
  }

}

void setVolume1()
{
  uint8_t tens = (uint8_t)volume1 / 10;
  uint8_t ones = (uint8_t)volume1 % 10;
  Wire.beginTransmission(address);
  Wire.write(0b10000000 | tens);
  Wire.write(0b10010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address);
  Wire.write(0b00100000 | tens);
  Wire.write(0b00110000 | ones);
  Wire.endTransmission();
  muted1 = false;
}

void setVolume2()
{
  uint8_t tens = (uint8_t)volume2 / 10;
  uint8_t ones = (uint8_t)volume2 % 10;
  Wire.beginTransmission(address);
  Wire.write(0b01000000 | tens);
  Wire.write(0b01010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address);
  Wire.write(0b01100000 | tens);
  Wire.write(0b01110000 | ones);
  Wire.endTransmission();
  muted2 = false;
}

void setVolume3()
{
  uint8_t tens = (uint8_t)volume3 / 10;
  uint8_t ones = (uint8_t)volume3 % 10;
  Wire.beginTransmission(address);
  Wire.write(0b00000000 | tens);
  Wire.write(0b00010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address);
  Wire.write(0b10100000 | tens);
  Wire.write(0b10110000 | ones);
  Wire.endTransmission();
  muted3 = false;
```

```
}

void setVolume4()
{
  uint8_t tens = (uint8_t)volume4 / 10;
  uint8_t ones = (uint8_t)volume4 % 10;
  Wire.beginTransmission(address2);
  Wire.write(0b10000000 | tens);
  Wire.write(0b10010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address2);
  Wire.write(0b00100000 | tens);
  Wire.write(0b00110000 | ones);
  Wire.endTransmission();
  muted4 = false;
}

void setVolume5()
{
  uint8_t tens = (uint8_t)volume5 / 10;
  uint8_t ones = (uint8_t)volume5 % 10;
  Wire.beginTransmission(address2);
  Wire.write(0b01000000 | tens);
  Wire.write(0b01010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address2);
  Wire.write(0b01100000 | tens);
  Wire.write(0b01110000 | ones);
  Wire.endTransmission();
  muted5 = false;
}

void setVolume6()
{
  uint8_t tens = (uint8_t)volume6 / 10;
  uint8_t ones = (uint8_t)volume6 % 10;
  Wire.beginTransmission(address2);
  Wire.write(0b00000000 | tens);
  Wire.write(0b00010000 | ones);
  Wire.endTransmission();
  Wire.beginTransmission(address2);
  Wire.write(0b10100000 | tens);
  Wire.write(0b10110000 | ones);
  Wire.endTransmission();
  muted6 = false;
}

void mute(String channel)
{
  if(channel.equals("0"))
  {
    if(muted == false)
    {
      Wire.beginTransmission(address);
      Wire.write(0b11010111);
      Wire.write(0b11101000);
      Wire.endTransmission();
      Wire.beginTransmission(address2);
      Wire.write(0b11010111);
      Wire.write(0b11101000);
      Wire.endTransmission();
      muted = true;
      muted1 = true;
```

```
    muted2 = true;
    muted3 = true;
    muted4 = true;
    muted5 = true;
    muted6 = true;
  }
  else
  {
    setVolume1();
    setVolume2();
    setVolume3();
    setVolume4();
    setVolume5();
    setVolume6();
    muted = false;
    muted1 = false;
    muted2 = false;
    muted3 = false;
    muted4 = false;
    muted5 = false;
    muted6 = false;
  }
}

if(channel.equals("1"))
{
  if(muted1 == false)
  {
    int temp = volume1;
    volume1 = 78;
    setVolume1();
    volume1 = temp;
    muted1 = true;
  }
  else
  {
    setVolume1();
    muted1 = false;
  }
}

if(channel.equals("2"))
{
  if(muted2 == false)
  {
    int temp = volume2;
    volume2 = 78;
    setVolume2();
    volume2 = temp;
    muted2 = true;
  }
  else
  {
    setVolume2();
    muted2 = false;
  }
}

if(channel.equals("3"))
{
  if(muted3 == false)
  {
    int temp = volume3;
```

```
      volume3 = 78;
      setVolume3();
      volume3 = temp;
      muted3 = true;
    }
    else
    {
      setVolume3();
      muted3 = false;
    }
  }

  if(channel.equals("4"))
  {
    if(muted4 == false)
    {
      int temp = volume4;
      volume4 = 78;
      setVolume4();
      volume4 = temp;
      muted4 = true;
    }
    else
    {
      setVolume4();
      muted4 = false;
    }
  }

  if(channel.equals("5"))
  {
    if(muted5 == false)
    {
      int temp = volume5;
      volume5 = 78;
      setVolume5();
      volume5 = temp;
      muted5 = true;
    }
    else
    {
      setVolume5();
      muted5 = false;
    }
  }

  if(channel.equals("6"))
  {
    if(muted6 == false)
    {
      int temp = volume6;
      volume6 = 78;
      setVolume6();
      volume6 = temp;
      muted6 = true;
    }
    else
    {
      setVolume6();
      muted6 = false;
    }
  }
}
```

The HTML File

```
<html>
<head>

<title>Kamdora</title> // Change this to whatever you want the name of your system to
be

<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

<script language='javascript' type='text/javascript'>
// Global variables.

var gServerIP = "192.168.1.199/";
var gScreenWidth;
var gScreenHeight;
var gSystemPowerStatus;
var gMutedAll;
var gMuted1;
var gVolume1;
var gMuted2;
var gVolume2;
var gMuted3;
var gVolume3;

function body_load() {
    gScreenWidth = window.innerWidth;
    gScreenHeight = window.innerHeight;

    setUiPosition(divMain, 0, 0, gScreenWidth, gScreenHeight);
       setUiPositionCanvas(canNavBar, 0, 0, gScreenWidth, 48);

       setUiPositionCanvas(canButtonOn, 0, 60, gScreenWidth / 2, 40);
       canButtonOn.OnDraw = canButtonOnDraw;
       addClickEvent(canButtonOn, canButtonOn_click);

       setUiPositionCanvas(canButtonOff, gScreenWidth / 2, 60, gScreenWidth / 2, 40);
       canButtonOff.OnDraw = canButtonOffDraw;
       addClickEvent(canButtonOff, canButtonOff_click);

       setUiPositionCanvas(canButtonMuteAll, 0, 112, gScreenWidth, 40);
       canButtonMuteAll.OnDraw = canButtonMuteAllDraw;
       addClickEvent(canButtonMuteAll, canButtonMuteAll_click);

       setUiPositionCanvas(canButtonVolume1Up, 0, 164, gScreenWidth / 6, 40);
       canButtonVolume1Up.OnDraw = canButtonVolume1UpDraw;
       addClickEvent(canButtonVolume1Up, canButtonVolume1Up_click);

       setUiPositionCanvas(canButtonVolume1Down, gScreenWidth - (gScreenWidth / 6),
164, gScreenWidth / 6, 40);
       canButtonVolume1Down.OnDraw = canButtonVolume1DownDraw;
       addClickEvent(canButtonVolume1Down, canButtonVolume1Down_click);

       setUiPositionCanvas(canButtonMute1, gScreenWidth / 6, 164, 5 * (gScreenWidth /
6), 40);
       canButtonMute1.OnDraw = canButtonMute1Draw;
       addClickEvent(canButtonMute1, canButtonMute1_click);
```

```
       setUiPositionCanvas(canButtonVolume2Up, 0, 216, gScreenWidth / 6, 40);
       canButtonVolume2Up.OnDraw = canButtonVolume2UpDraw;
       addClickEvent(canButtonVolume2Up, canButtonVolume2Up_click);

       setUiPositionCanvas(canButtonVolume2Down, gScreenWidth - (gScreenWidth / 6),
216, gScreenWidth / 6, 40);
       canButtonVolume2Down.OnDraw = canButtonVolume2DownDraw;
       addClickEvent(canButtonVolume2Down, canButtonVolume2Down_click);

       setUiPositionCanvas(canButtonMute2, gScreenWidth / 6, 216, 5 * (gScreenWidth /
6), 40);
       canButtonMute2.OnDraw = canButtonMute2Draw;
       addClickEvent(canButtonMute2, canButtonMute2_click);

       setUiPositionCanvas(canButtonVolume3Up, 0, 268, gScreenWidth / 6, 40);
       canButtonVolume3Up.OnDraw = canButtonVolume3UpDraw;
       addClickEvent(canButtonVolume3Up, canButtonVolume3Up_click);

       setUiPositionCanvas(canButtonVolume3Down, gScreenWidth - (gScreenWidth / 6),
268, gScreenWidth / 6, 40);
       canButtonVolume3Down.OnDraw = canButtonVolume3DownDraw;
       addClickEvent(canButtonVolume3Down, canButtonVolume3Down_click);

       setUiPositionCanvas(canButtonMute3, gScreenWidth / 6, 268, 5 * (gScreenWidth /
6), 40);
       canButtonMute3.OnDraw = canButtonMute3Draw;
       addClickEvent(canButtonMute3, canButtonMute3_click);

    canNavBarDraw();
       getVars();
}

function canNavBarDraw() {
    var canvas2d = canNavBar.getContext("2d");

    canvas2d.fillStyle = "#0858a8";
    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "bold 22px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("Kamdora", gScreenWidth / 2, 24);
}

function canButtonOnDraw(isPressed) {
    var canvas2d = canButtonOn.getContext("2d");

    if (isPressed == true) {
        //canvas2d.fillStyle = "#005500";
             gSystemPowerStatus = true;
             canButtonOffDraw(false);
    }
    //else {
        //canvas2d.fillStyle = "#007700";
    //}
       if(gSystemPowerStatus == true){
```

```
            canvas2d.fillStyle = "#00aa00";
      }
      if(gSystemPowerStatus == false){
            canvas2d.fillStyle = "#004400";
      }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("System On", gScreenWidth / 4, 20);
}

function canButtonOn_click() {
      var response = httpGet("?on");
      getVars();
}

function canButtonOffDraw(isPressed) {
    var canvas2d = canButtonOff.getContext("2d");

    if (isPressed == true) {
        //canvas2d.fillStyle = "#880000";
            gSystemPowerStatus = false;
            canButtonOnDraw(false);
    }
   // else {
     //   canvas2d.fillStyle = "#aa0000";
    //}
      if(gSystemPowerStatus == false){
            canvas2d.fillStyle = "#cc0000";
      }
      else{
            canvas2d.fillStyle = "#550000";
      }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("System Off", gScreenWidth / 4, 20);
}

function canButtonOff_click() {
      var response = httpGet("?off");
      getVars();
}
```

```
function canButtonMuteAllDraw(isPressed) {
    var canvas2d = canButtonMuteAll.getContext("2d");

    if (isPressed == true) {
              if(gMutedAll == false)
              {      gMutedAll = true;
              }
              if(gMutedAll == true)
                     {gMutedAll = false;}
    }
       if(gMutedAll == true)
       {
              canvas2d.fillStyle = "#aa0000";
       }
       else
       {
              canvas2d.fillStyle = "#00aa00";
              }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
       if(gMutedAll == false) {
              canvas2d.fillText("MUTE ALL", gScreenWidth / 2, 20);
       }
       if(gMutedAll == true) {
              canvas2d.fillText("MUTE ALL [MUTED]", gScreenWidth / 2, 20);
       }
}

function canButtonMuteAll_click() {
       var response = httpGet("?m0");
       getVars();
}

function getVars() {
       var values = httpGet("?getVars");
       var varArray = values.split("*");
       gMutedAll = varArray[1];
       gMuted1 = varArray[2];
       gVolume1 = varArray[3];
       gMuted2 = varArray[4];
       gVolume2 = varArray[5];
       gMuted3 = varArray[6];
       gVolume3 = varArray[7];
       gSystemPowerStatus = varArray[14];
       loadButtons();
}

function loadButtons()
{
       canButtonOnDraw(false);
       canButtonOffDraw(false);
```

```
       canButtonMuteAllDraw(false);
       canButtonVolume1UpDraw(false);
       canButtonVolume1DownDraw(false);
       canButtonMute1Draw(false);
       canButtonVolume2UpDraw(false);
       canButtonVolume2DownDraw(false);
       canButtonMute2Draw(false);
       canButtonVolume3UpDraw(false);
       canButtonVolume3DownDraw(false);
       canButtonMute3Draw(false);
}

function canButtonVolume1UpDraw(isPressed) {
    var canvas2d = canButtonVolume1Up.getContext("2d");

    if (isPressed == true) {
             canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
             canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2191", gScreenWidth / 12, 20);
}

function canButtonVolume1Up_click() {
       if(gMuted1 == false) {
             if(gVolume1 > 1)
             {
                    gVolume1--;
                    gVolume1--;
             }
             var response = httpGet("?v1" + gVolume1);
             canButtonMute1Draw();
       }
       getVars();
}

function canButtonVolume1DownDraw(isPressed) {
    var canvas2d = canButtonVolume1Down.getContext("2d");

    if (isPressed == true) {
             canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
             canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
```

```
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2193", gScreenWidth / 12, 20);
}

function canButtonVolume1Down_click() {
       if(gMuted1 == false) {
              if(gVolume1 < 77)
              {
                     gVolume1++;
                     gVolume1++;
              }
              var response = httpGet("?v1" + gVolume1);
              canButtonMute1Draw();
       }
       getVars();
}

function canButtonMute1Draw(isPressed) {
    var canvas2d = canButtonMute1.getContext("2d");

    if (isPressed == true) {
              if(gMuted1 == false)
              {      gMuted1 = true;}
              if(gMuted1 == true)
                     {gMuted1 = false;}
    }
       if(gMuted1 == true)
       {
              canvas2d.fillStyle = "#aa0000";
              }
       if(gMuted1 == false)
       {
              canvas2d.fillStyle = "#00aa00";
              }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
       if(gMuted1 == true) {
              canvas2d.fillText("Channel 1 Mute [MUTED]", gScreenWidth / 3, 20);
       }
    if(gMuted1 == false) {
              canvas2d.fillText("Channel 1 Mute [" + gVolume1 + "]", gScreenWidth / 3,
20);
       }
}
```

```
function canButtonMute1_click() {
       var response = httpGet("?m1");
       getVars();
}

function canButtonVolume2UpDraw(isPressed) {
    var canvas2d = canButtonVolume2Up.getContext("2d");

    if (isPressed == true) {
              canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
              canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2191", gScreenWidth / 12, 20);
}

function canButtonVolume2Up_click() {
       if(gMuted2 == false) {
              if(gVolume2 > 1)
              {
                     gVolume2--;
                     gVolume2--;
              }
              var response = httpGet("?v2" + gVolume2);
              canButtonMute2Draw();
       }
       getVars();
}

function canButtonVolume2DownDraw(isPressed) {
    var canvas2d = canButtonVolume2Down.getContext("2d");

    if (isPressed == true) {
              canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
              canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;
```

```
    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2193", gScreenWidth / 12, 20);
}

function canButtonVolume2Down_click() {
       if(gMuted2 == false) {
              if(gVolume2 < 77)
              {
                     gVolume2++;
                     gVolume2++;
              }
              var response = httpGet("?v2" + gVolume2);
              canButtonMute2Draw();
       }
       getVars();
}

function canButtonMute2Draw(isPressed) {
    var canvas2d = canButtonMute2.getContext("2d");

    if (isPressed == true) {
              if(gMuted2 == false)
              {      gMuted2 = true;}
              if(gMuted2 == true)
                     {gMuted2 = false;}
    }
       if(gMuted2 == true)
       {
              canvas2d.fillStyle = "#aa0000";
              }
       if(gMuted2 == false)
       {
              canvas2d.fillStyle = "#00aa00";
              }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
       if(gMuted2 == true) {
              canvas2d.fillText("Channel 2 Mute [MUTED]", gScreenWidth / 3, 20);
       }
    if(gMuted2 == false) {
              canvas2d.fillText("Channel 2 Mute [" + gVolume2 + "]", gScreenWidth / 3,
20);
       }
}

function canButtonMute2_click() {
       var response = httpGet("?m2");
       getVars();
}
```

```
function canButtonVolume3UpDraw(isPressed) {
    var canvas2d = canButtonVolume3Up.getContext("2d");

    if (isPressed == true) {
              canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
              canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2191", gScreenWidth / 12, 20);
}

function canButtonVolume3Up_click() {
       if(gMuted3 == false) {
              if(gVolume3 > 1)
              {
                     gVolume3--;
                     gVolume3--;
              }
              var response = httpGet("?v3" + gVolume3);
              canButtonMute3Draw();
       }
       getVars();
}

function canButtonVolume3DownDraw(isPressed) {
    var canvas2d = canButtonVolume3Down.getContext("2d");

    if (isPressed == true) {
              canvas2d.fillStyle = "#000055";
    }
       if(isPressed == false) {
              canvas2d.fillStyle = "#0000aa";
       }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
    canvas2d.fillText("\u2193", gScreenWidth / 12, 20);
}
```

```
function canButtonVolume3Down_click() {
       if(gMuted3 == false) {
              if(gVolume3 < 77)
              {
                     gVolume3++;
                     gVolume3++;
              }
              var response = httpGet("?v3" + gVolume3);
              canButtonMute3Draw();
       }
       getVars();
}

function canButtonMute3Draw(isPressed) {
    var canvas2d = canButtonMute3.getContext("2d");

    if (isPressed == true) {
              if(gMuted3 == false)
              {      gMuted3 = true;}
              if(gMuted3 == true)
                     {gMuted3 = false;}
    }
       if(gMuted3 == true)
       {
              canvas2d.fillStyle = "#aa0000";
              }
       if(gMuted3 == false)
       {
              canvas2d.fillStyle = "#00aa00";
              }

    canvas2d.fillRect(0, 0, gScreenWidth, 48);

    canvas2d.fillStyle = "white";
    canvas2d.textBaseline = "middle";

    canvas2d.shadowColor = "grey";
    canvas2d.shadowOffsetX = 1;
    canvas2d.shadowOffsetY = 1;
    canvas2d.shadowBlur = 1;

    canvas2d.font = "20px Helvetica";
    canvas2d.textAlign = "center";
       if(gMuted3 == true) {
              canvas2d.fillText("Channel 3 Mute [MUTED]", gScreenWidth / 3, 20);
       }
    if(gMuted3 == false) {
              canvas2d.fillText("Channel 3 Mute [" + gVolume3 + "]", gScreenWidth / 3,
20);
       }
}

function canButtonMute3_click() {
       var response = httpGet("?m3");
       getVars();
}
// General-purpose functions.

function httpGet(cmd) {

       var url = gServerIP + cmd;

       var xmlHttp = null;
```

```
    xmlHttp = new XMLHttpRequest();
    xmlHttp.open("GET", url, false);
    xmlHttp.send(null);

    return xmlHttp.responseText;
}

function body_touchMove(event) {
    event.preventDefault(); // Prevents elastic scrolling.
}

function isTouchDevice() {
    return "ontouchstart" in window;
}

function addClickEvent(object, eventFunction) {

    object.OnClick = eventFunction;

    if (isTouchDevice() === true) {
        object.ontouchstart = clickStartTouch;
        object.ontouchend = clickEnd;
    }
    else {
        object.onmousedown = clickStartMouse;
        object.onmouseup = clickEnd;
    }

    object.OnDraw(false);

    function clickStartTouch(event) {
        object.OnDraw(true);
        object.ClickX = event.touches[0].clientX;
        object.ClickY = event.touches[0].clientY;
    }

    function clickStartMouse(event) {
        object.OnDraw(true);
        object.ClickX = event.clientX;
        object.ClickY = event.clientY;
    }

    function clickEnd(event) {
        object.OnDraw(false);
        object.OnClick(object);
    }
}

function addLongPressEvent(object, eventFunction) {

    object.OnLongPress = eventFunction;

    if (isTouchDevice() === true) {
        object.ontouchstart = longPressStart;
        object.ontouchmove = longPressMove;
        object.ontouchend = longPressEnd;
    }
    else {
        object.onmousedown = longPressStart;
        object.onmousemove = longPressMove;
        object.onmouseup = longPressEnd;
    }
```

```
    function longPressStart() {
        object.TimeoutID = setTimeout(longPressTimeout, 1000);
    }

    function longPressMove() {
        clearTimeout(object.TimeoutID);
    }

    function longPressEnd() {
        clearTimeout(object.TimeoutID);
    }

    function longPressTimeout() {
        object.OnLongPress(object);
    }
}

function setUiPosition(object, left, top, width, height) {
    object.style.position = "absolute";
    object.style.left = left;
    object.style.top = top;
    object.style.width = width;
    object.style.height = height;
}

function setUiPositionCanvas(object, left, top, width, height) {
    object.width = width;
    object.height = height;

    object.style.position = "absolute";
    object.style.left = left;
    object.style.top = top;
}

function localStorageGet(keyName, dfltValue) {

    if (localStorage.getItem(keyName) == undefined) {
        return dfltValue;
    }

    return localStorage.getItem(keyName);
}
</script>

<style type="text/css">

body {
    margin:0px;
    padding:0px;
    background:red;
    overflow:hidden;
    -webkit-text-size-adjust:none;
    -webkit-user-select:none;
    -webkit-touch-callout:none;
}

#divMain {
    position:absolute;
    background-color:#dddddd;
}

</style>
```

```
</head>

<body
    onload="body_load();"
    ontouchmove="body_touchMove(event);"
>

<div id="divMain">

    <canvas id="canNavBar">
    </canvas>

    <canvas id="canButtonOn">
    </canvas>

    <canvas id="canButtonOff">
    </canvas>

       <canvas id ="canButtonMuteAll">
       </canvas>

       <canvas id ="canButtonMute1">
       </canvas>

       <canvas id ="canButtonVolume1Up">
       </canvas>

       <canvas id ="canButtonVolume1Down">
       </canvas>

       <canvas id ="canButtonMute2">
       </canvas>

       <canvas id ="canButtonVolume2Up">
       </canvas>

       <canvas id ="canButtonVolume2Down">
       </canvas>

       <canvas id ="canButtonMute3">
       </canvas>

       <canvas id ="canButtonVolume3Up">
       </canvas>

       <canvas id ="canButtonVolume3Down">
       </canvas>


</div>

</body>
</html>
```