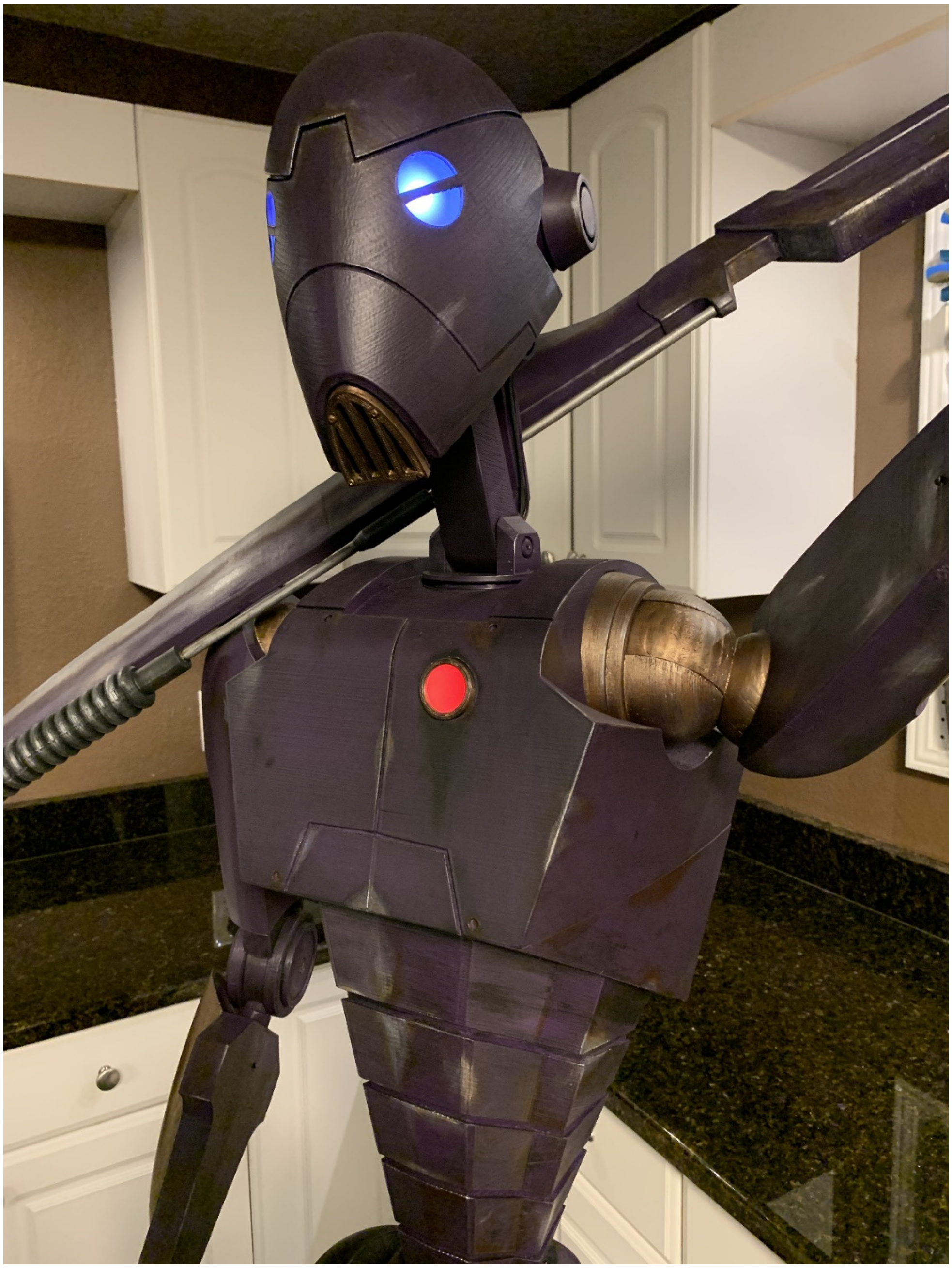
# Commando Droid Electrical Implementation

**May 2020**



# Commando Droid Electrical Implementation

# 1 DISCLAIMER

You are hereby authorized to view, copy, print, and distribute the materials from this document or the attached drawings subject to the following conditions:

- No liability in respect of any of the contents of such publications or use thereof is accepted and no warranties expressed or implied are made in relation thereto.

**Disclaimer**

ANY INFORMATION CONTAINED IN THIS DOCUMENT OR THE ATTACHED DRAWINGS ARE PROVIDED [AS IS] FOR YOUR INFORMATIONAL PURPOSES ONLY, WITHOUT WARRANTY OF ANY KIND, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THE AUTHOR OF THIS DOCUMENT AND ATTACHED DRAWINGS WILL IN NO EVENT BE LIABLE FOR ANY DIRECT, INDIRECT OR PUNITIVE DAMAGE OF ANY KIND THAT RESULTS FROM THE USE OF OR INABILITY TO USE THIS DOCUMENT OR THE ATTACHED DRAWINGS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, GOODWILL, OR BUSINESS INTERRUPTION. THE AUTHOR OFFERS NO ASSURANCES, WARRANTIES OR GUARANTEES AS TO THE VALIDITY OR COMPLETENESS OF THIS INFORMATION. INFORMATION WILL BE CHANGED, UPDATED AND DELETED WITHOUT NOTICE.

## 2 Overview

The Commando Droid .stl files were purchased from the following link:
https://www.etsy.com/listing/781395972/spacebobs-commando-droid-printable-files?ref=shop_home_active_6&crt=1

The intent of this document is to provide information on the methodology I used to enhance the commando droid by using WS2812B LEDSs and adding audio. The focus of this document is on the electrical components used and a brief summary of what was changed to incorporate the electronics.

The project materials utilized are summarized below and a detailed list of the components for purchasing the materials is listed in section 6 of this document.

- (1) Arduino Nano
- (4) WS2812B LEDS
- (1) Panel Mount 2.1mm DC barrel jack
- (1) 12 VDC Wall Power Adapter (2000mA) – Connector 2.1mm plug
- (1) Step Down Regulator 12V to 5V
- (1) Electronics123.com, Inc. 4 Buttons Triggered MP3 Player Board
- (1) SparkFun Sound Detector
- (1) Speaker
- (1) Wireless RF Switch Long Range DC 12V 4CH Channel Wireless Remote

You should test that you have everything wired correctly by running the code via the Arduino Nano on a test bench prior to installing the hardware in the Commando droid. Just hook up everything in a temporary fashion on a test bench to make sure your solder joints are good and everything is functional. This initial troubleshooting step is much easier to do prior to the actual installation.

# 3 3D Printed Files

## 3.1 Power Connection Box

I made a power box that mounts to the back of the foot as shown below. The box holds the 2.1 mm DC barrel jack.

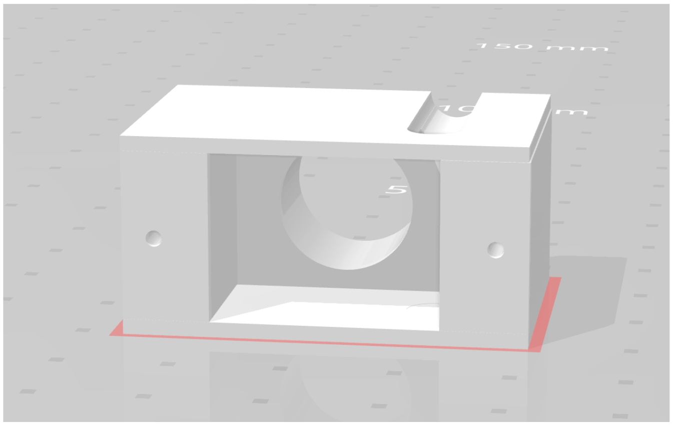


Figure 3-1 DC Barrel Jack Holder

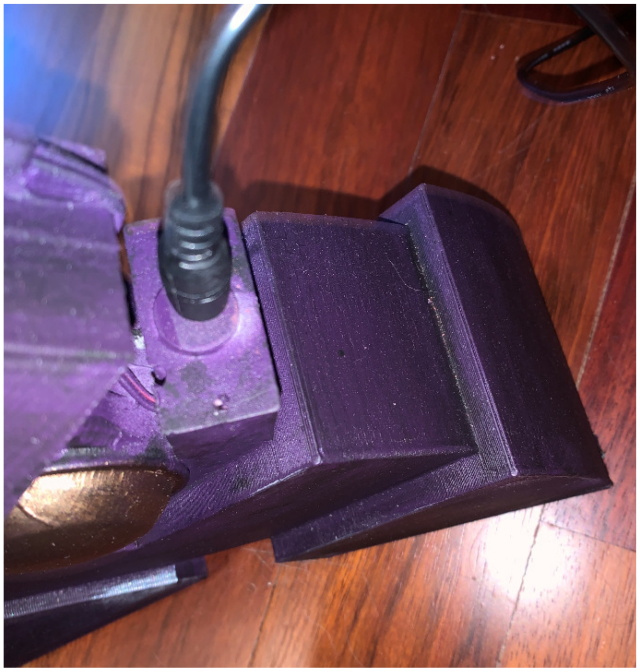


Figure 3-2 DC Barrel Jack Holder Installed

## 3.2 Head Cap Modification

I modified the head cap to allow the remote-control box and audio card to be installed. The recess shown is to allow the remote-control box to sit flat.

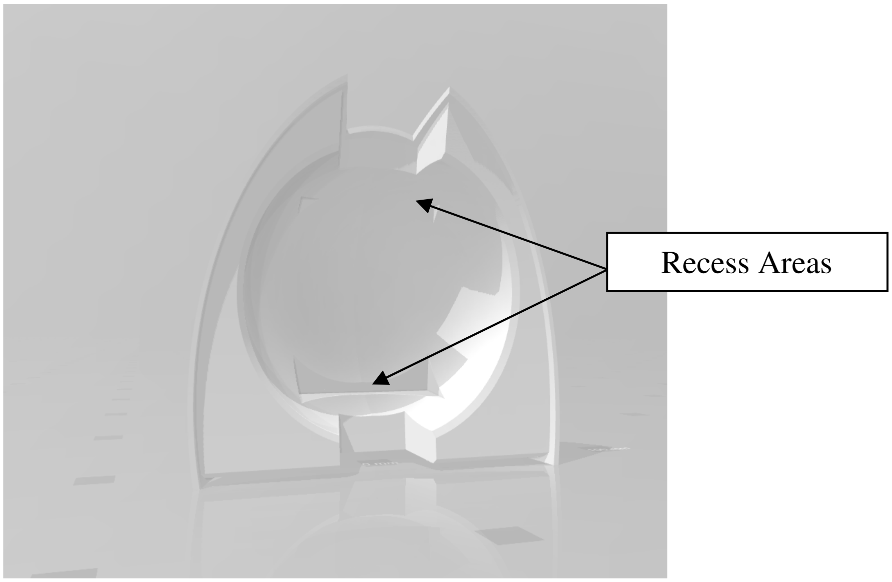


Figure 3-3 Modified Head Cap

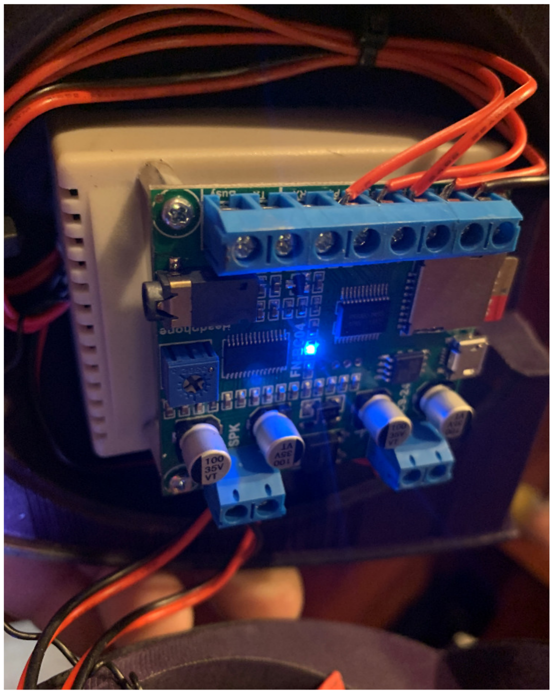


Figure 3-4 Remote Box and Audio Installed in Head Cap

## 3.3 Head Modification

I modified the head as shown below. The modification required using Windows 10 3D builder to carve out areas on the existing head .stl file. I ended up making the speaker grill holes larger than shown below using a drill since they were too small once the head was printed.

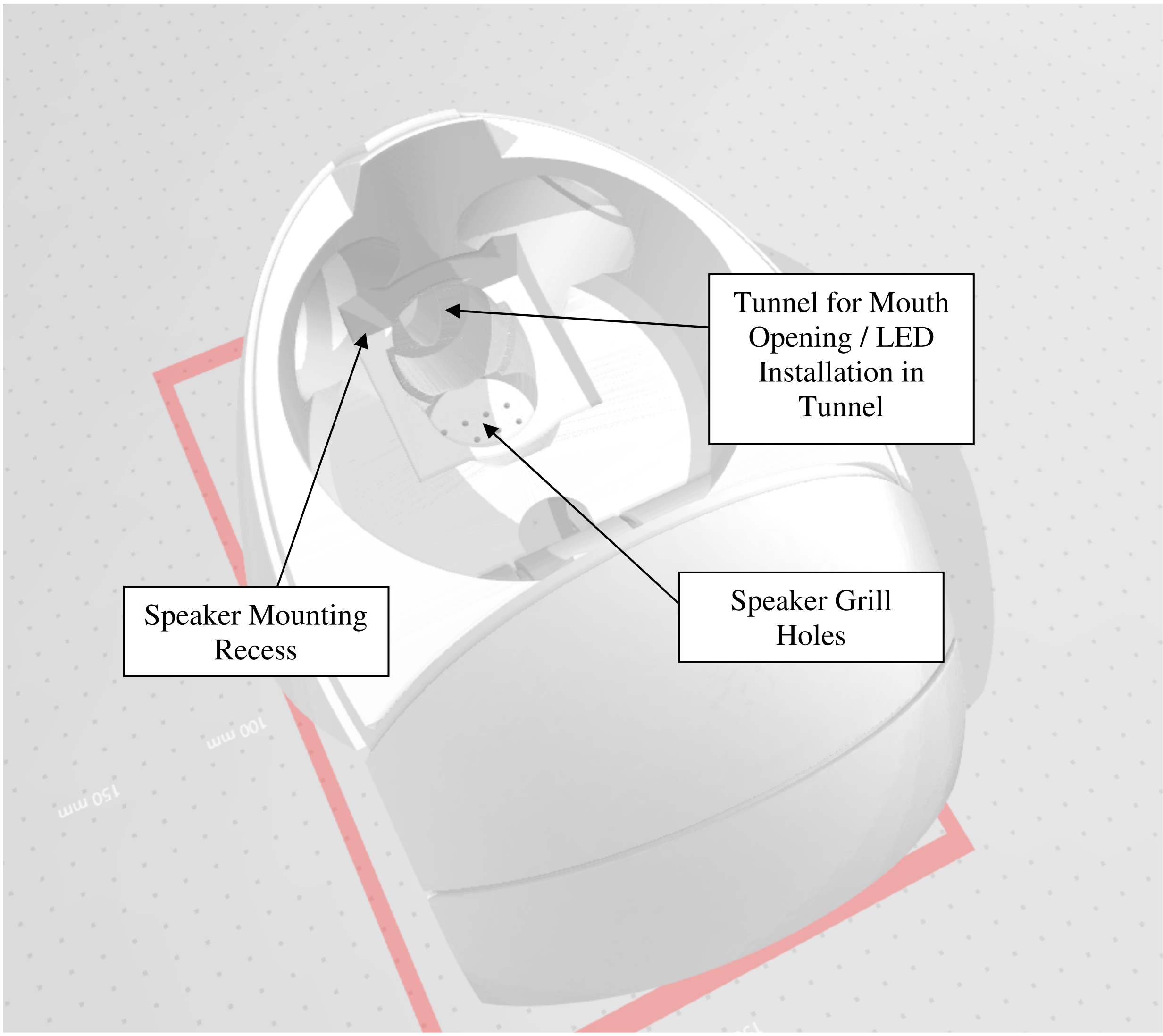


Figure 3-5 Head Modification

## 3.4 Speaker Grill

I created a small speaker grill that is installed on the bottom of the head as shown.

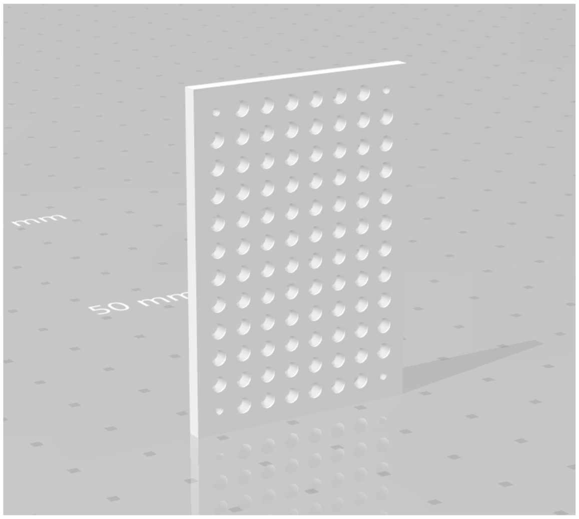


Figure 3-6 Speaker Grill

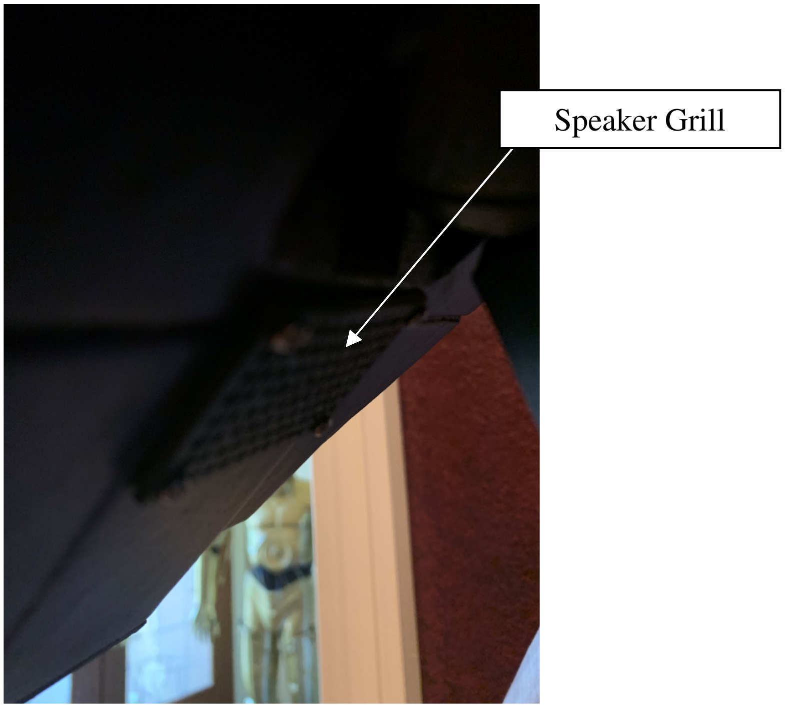


Figure 3-7 Speaker Grill Installed

# 4 Paint

## 4.1 Body Color

I wanted a purple body color for the Commando Droid. I first tried Montana GOLD Black Purple spray paint. This paint is an acrylic and once you do a black wash, it comes out too black and does not take a weathering wash well.

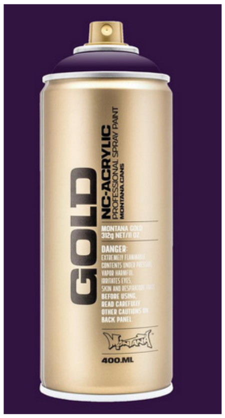


Figure 4-1 Montana GOLD Black Purple Paint

When the Montana paint was not acceptable, I ended up using Rust-Oleum Gloss Purple paint.

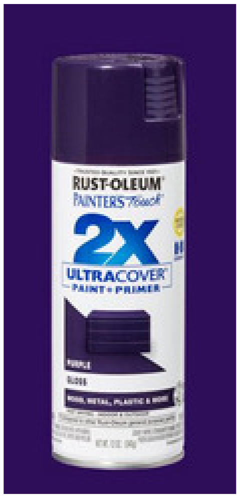


Figure 4-2 Rust-Oleum Gloss Purple Paint

When I do weathering, I prefer to use a satin paint then black wash over it. Gloss paint does not provide a good weathering effect when black washing. The key to using gloss paint is toe mute it down. To do this I sprayed over the gloss paint with Rust-Oleum Matte Clear spray paint. You must let the gloss dry for at least a few days and lightly mist over the gloss paint. If you spray the matte paint too soon or apply it too think, the gloss paint will crack and bubble.

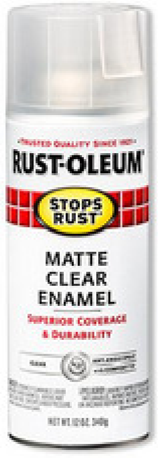


Figure 4-3 Rust-Oleum Matte Clear Paint

Once you have sprayed the gloss and then the matte paint, you are ready to black wash. I just put a blob of black acrylic paint on a disposable paper plate and then take my chip brush and dip it in water and load the brush up with the acrylic paint. Slop the paint all over the part and then use paper towel to wipe off the black acrylic paint to achieve the desired for effect. This provides a nice base to add rust and silver scrape marks to your project. I used the silver rub and buff to create many of the weather effects. The key is to put the rub and buff on your droid as dry as possible and lightly rub. I found that dry brushing a little dark brown around the silver edges adds to the weathered look.

For the shoulder joints, knee joints, mouth grill and areas on the outer arm sections, I used Art alchemy Bronze Age metallic wax I purchased on Amazon.

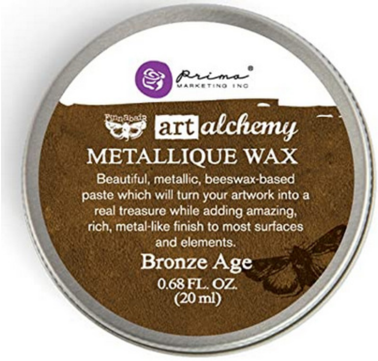

Prima MARKETING INC
art alchemy
METALLIQUE WAX
Beautiful, metallic, beeswax-based paste which will turn your artwork into a real treasure while adding amazing, rich, metal-like finish to most surfaces and elements.
Bronze Age
0.68 FL. OZ.
(20 ml)

Figure 4-4 Bronze Age Metallic Wax

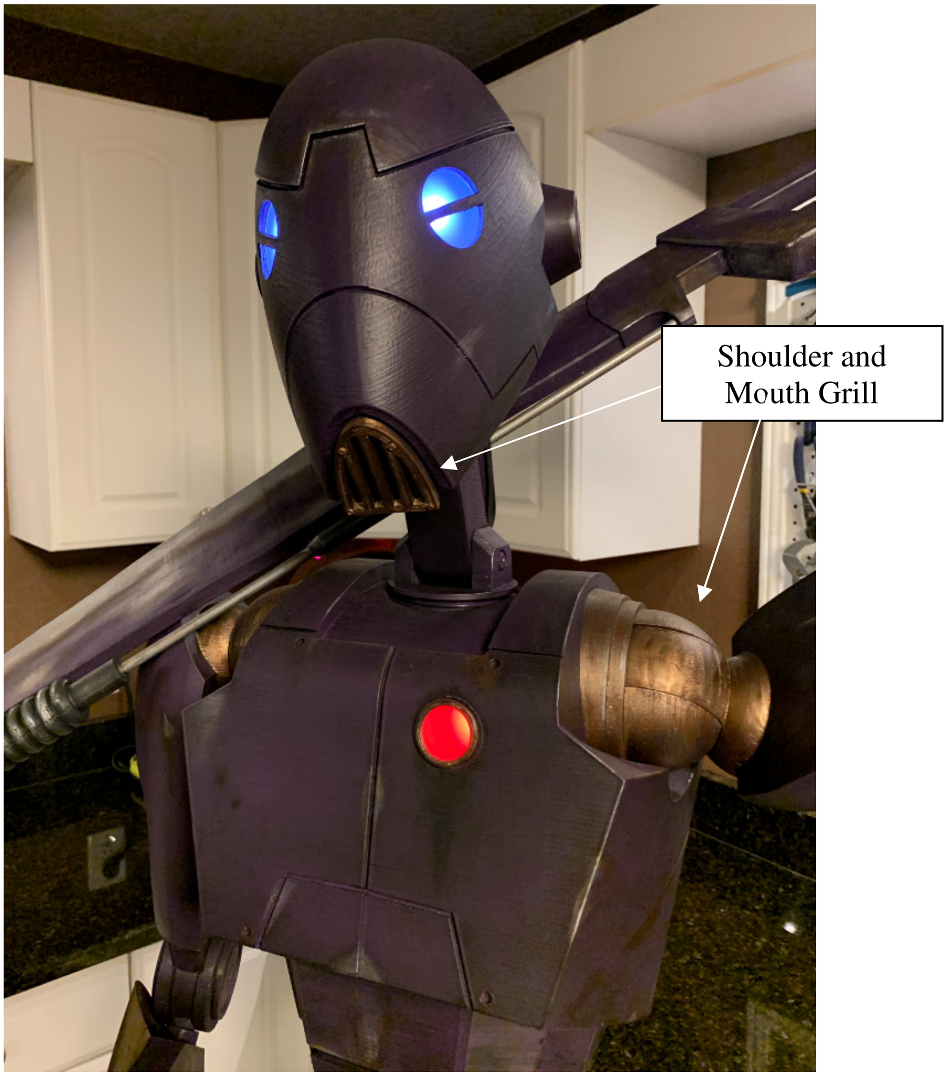

Shoulder and Mouth Grill

Figure 4-5 Bronze Age Location

## 5 Arduino Nano Controller

The Arduino Nano shown below provides the control of the WS2812B LEDS and has an input that is controlled by the SparkFun sound detector. The Arduino is mounted in the head of the commando droid (Pin Side Up) so we can easily make the power and LED connections. The Arduino programming software is free and can be obtained from the following link: The wiring diagram will provide you with all of the connections required.

[https://www.arduino.cc/en/main/software](https://www.arduino.cc/en/main/software)

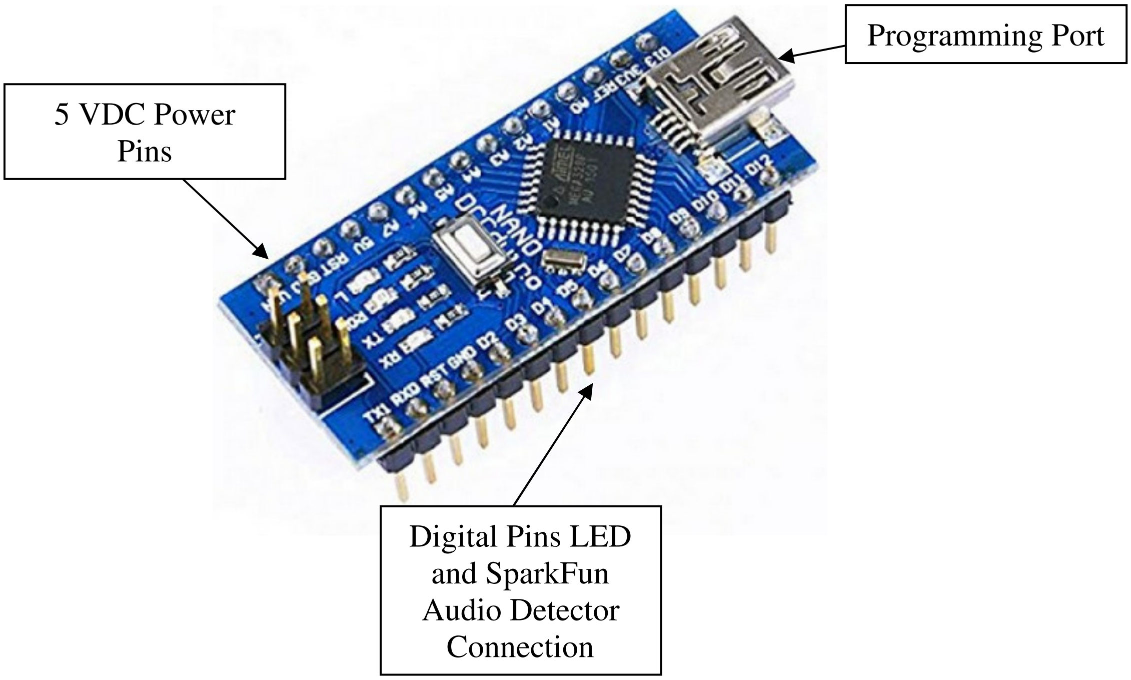


Figure 5-1 Arduino Nano

## 5.1 Arduino Programming

The Arduino Nano code that I am running in the Commando droid is listed below.

```
// Commando Droid Droid WS2812B and Voice
//John Guarnero 2020

#include <Adafruit_NeoPixel.h>

//Define Arduino Pins to control NeoPixels / WS2812B LEDS
#define Pin_Mouth 3 //Pin 3 is the Mouth
#define Pin_Eyes 4 //Pin 4 is the Eyes
#define Pin_Chest 5 //Pin 5 is the Chest

// Number of NeoPixels / WS2812B LEDS attached to each pin
#define NumPixels_Mouth 1  //Neopixel Mini Button 1 Neopixel
#define NumPixels_Chest 1  //Neopixel Mini Button 1 Neopixel
#define NumPixels_Eyes 2   //Neopixel Mini Button 1 Neopixel

// When setting up the NeoPixel / WS2812B LEDS library, we tell it how many
pixels,
// and which pin to use to send signals.
Adafruit_NeoPixel Mouth(NumPixels_Mouth, Pin_Mouth, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel Eyes(NumPixels_Eyes, Pin_Eyes, NEO_RGBW +
NEO_KHZ800);
Adafruit_NeoPixel Chest(NumPixels_Chest, Pin_Chest, NEO_RGBW +
NEO_KHZ800);

//Define Variables used in the program
int whitecolor = 50;  // White color intensiy (0-250 allowed)
int pin_val;
int BlinkRate;
int Chest_Pixels = 0;
int Chest_Fade_Up = 0;
int Chest_Fade_Down = 1;
unsigned long previousMillis = 0;
unsigned long previousMillischest = 0;


void setup()

{
 Mouth.begin();
 Mouth.clear();
 Eyes.begin();
 Eyes.clear();
 Chest.begin();
 Chest.clear();

}

//Program runs code below - Void Loop
void loop()
{

//Eyes
   BlinkRate = random(6000, 12000);  // BlinkRate will randomly become active
between 6 seconds and 12 seconds

  unsigned long currentMillis = millis();   //millis is used to avoid using a delay
command in the code
 if (currentMillis - previousMillis > 50) {
   Eyes.setPixelColor(0, Eyes.Color(whitecolor, whitecolor, whitecolor, whitecolor));
   Eyes.setPixelColor(1, Eyes.Color(whitecolor, whitecolor, whitecolor, whitecolor));
   Eyes.show();  //Update the Eye NeoPixels
  } //IF statement end bracket

   if (currentMillis - previousMillis >= BlinkRate) { // Blink the Eyes off
    previousMillis = currentMillis;
   Eyes.setPixelColor(0, Eyes.Color(0, 0, 0, 0));   //Turn off Eye Leds
   Eyes.setPixelColor(1, Eyes.Color(0, 0, 0, 0));   //Turn off Eye Leds
   Eyes.show();  //Update the Eye NeoPixels
  } //IF statement end bracket

//Chest
   unsigned long currentMillischest = millis();

   if (currentMillischest - previousMillischest >50) {
    previousMillischest = currentMillischest;

//Chest LED is set to Red color that fades up and down
//Red Color Fade Up
 if (Chest_Fade_Up == 1) {
  Chest_Pixels = Chest_Pixels + 10;
  if (Chest_Pixels > 250) {
  Chest_Fade_Up = 0;
  Chest_Fade_Down = 1;
  } //IF statement end bracket
  } //IF statement end bracket

//Red Color Fade Down
if (Chest_Fade_Down == 1) {
  Chest_Pixels = Chest_Pixels - 10;
  if (Chest_Pixels < 20) {
  Chest_Fade_Up = 1;
  Chest_Fade_Down = 0;
  } //IF statement end bracket
  } //IF statement end bracket

   Chest.setPixelColor(0, Chest.Color(0,Chest_Pixels, 0, 0)); //The Chest just flashed
random colors
   Chest.show();
    } //IF statement end bracket


//Mouth - Set Neopixel intesity based on Sparkfun gate input
  pinMode(6, INPUT);
  pin_val = digitalRead(6);

 if (pin_val == 1) { //Mouth LED On
  Mouth.setPixelColor(0, 250,0,0); //The Mouth Green, Red, Blue, White
  Mouth.show();
  } //IF statement end bracket

   if (pin_val == 0) { //Mouth LED Off
  Mouth.setPixelColor(0, 0,0,0); //The Mouth Green, Red, Blue, White
  Mouth.show();
  } //IF statement end bracket

  } // End of Program Void Loop end bracket
```

## 6 Parts List

The following list details all of the electrical components used in the system.

| Description | Part Number | QTY | Purchased From |
|---|---|---|---|
| Panel Mount 2.1mm DC barrel jack | ID:1612 | 1 | Adafruit.com |
| BTF-LIGHTING 100 pcs WS2812B LED chips With PCB Heatsink (10mm3mm) WS2811 IC Built-in 5050 SMD RGB DC5V | - | 1 | amazon.com |
| 65.6ft Extension Cable Wire Cord JACKYLED 20M 22AWG Wire Cord for Led Strips Single Color 3528 5050 | - | 1 | amazon.com |
| Cylewet 2Pcs 2inch 8Ohm 12W Midrange Speaker Low Frequency Loudspeaker Low | - | 1 | amazon.com |
| SparkFun Sound Detector Audio sensing breakout Three different outputs Audio Presence of sound Binary indication Amplitude Analog representation VCC: 3.5V to 5.5V Ideal voltage is 5V | - | 1 | amazon.com |
| Electronics123.com, Inc. 4 Buttons Triggered MP3 Player Board with 10W Amplifier and Terminal Blocks | - | 1 | amazon.com |
| Wireless RF Switch Long Range DC 12V 4CH Channel Wireless Remote | - | 1 | amazon.com |

| **Description** | **Part Number** | **QTY** | **Purchased From** |
|---|---|---|---|
| BINZET DC Converter Step Down Regulator 5V Regulated Power Supplies Transformer Converter (5V 3A 15W) | - | 1 | amazon.com |
| TMEZON 12 Volt 2A Power Adapter Supply AC to DC 2.1mm X 5.5mm Plug 12v 2 Amp Power Supply Wall Plug Extra Long 8 Foot Cord | - | 1 | amazon.com |
| Arduino Nano | - | 1 | amazon.com |

## 7 Electrical Summary and Schematics

The wiring schematics are provided at the end of this document. A brief summary of the electrical circuit is as follow:

**Audio Playback**

I used a (Electronics123.com, Inc. 4 Buttons Triggered MP3 Player Board with 10W Amplifier and Terminal Blocks) board that I purchased on Amazon to playback the audio files. The board uses a microSD card that you place your mp3 or wav files on. The electronics123.com website has very good instructions for using this module. The module has four (4) audio file triggers that will play an audio file when two terminals are shorted. The module can also receive a serial string to trigger several different audio tracks. For the purpose of the Commando droid, I am using the four (4) individual triggers. I will be using the serial interface for the Smelter droid since he will tell the time on the hour and requires the ability to playback several audio files.

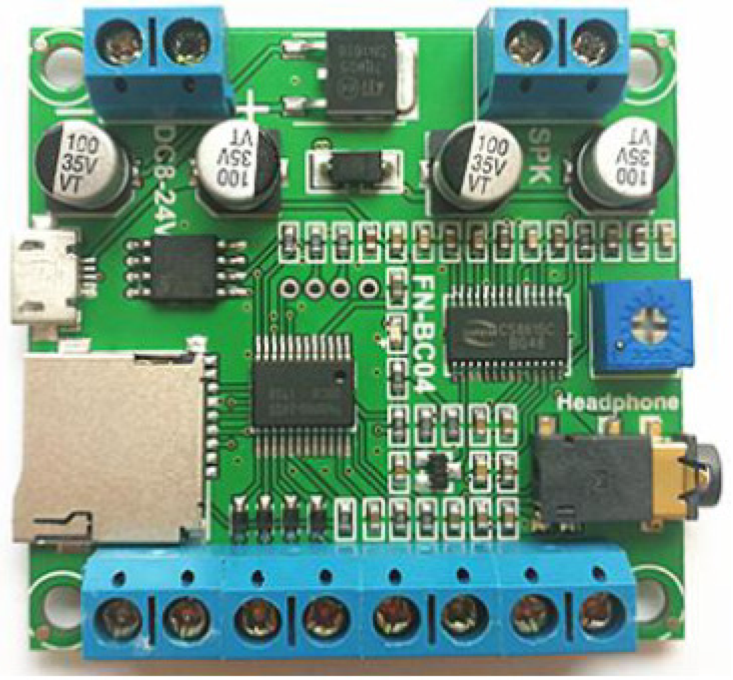


Figure 7-1 Electronics123.com, Inc. 4 Buttons Triggered MP3 Player Board

**Audio Triggering – Remote Control**

I used a (Wireless RF Switch Long Range DC 12V 4CH Channel Wireless Remote) that I purchased on Amazon to trigger the audio playback board. The remote switch has four relays and when a remote-control pushbutton is pressed, the corresponding relay on the board will activate. The four (4) relay contact are wired to the audio playback board. All of this is done without the Arduino involved. You can use most any remote switch since they are pretty much all the same.

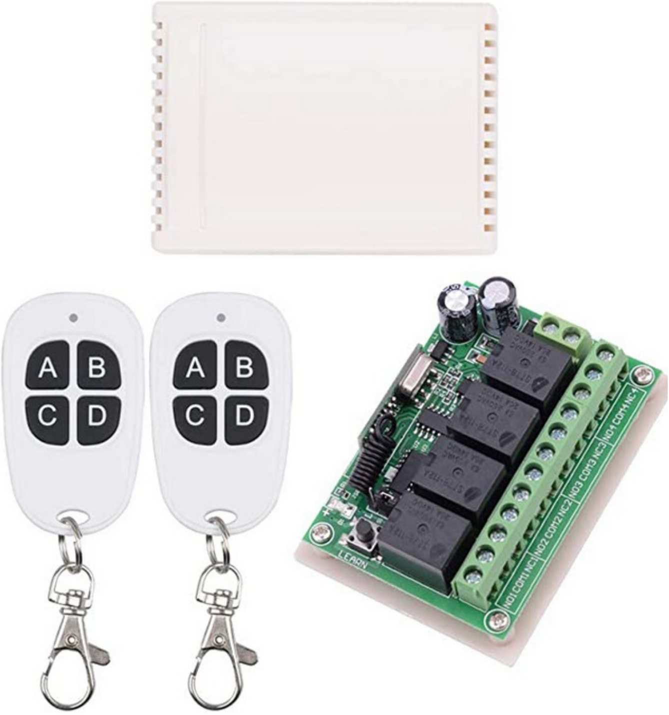


Figure 7-2 Four (4) Channel Remote Control

## Audio Detection

I used a (SparkFun Sound Detector Audio sensing breakout) that I purchased on Amazon to detect audio level and provide a signal to the Arduino Nano when the decibel threshold is reached. I placed this board near the speaker inside the head. The board has a small microphone to detect audio. The gain for the audio detection can be changed and I reduced the gain around 70% using a resistor that can be soldered on the board. The instructions from SparkFun provides a table of gain settings and suggested resistor values. Their manual is quite good.

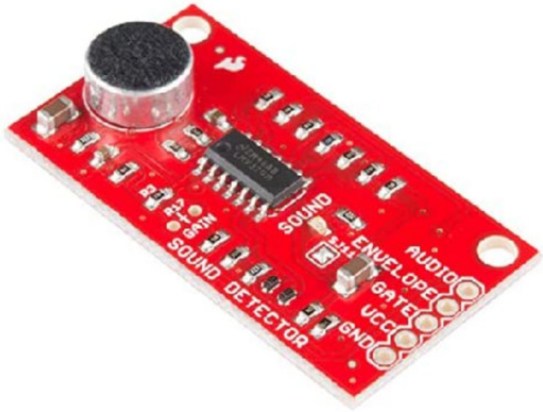


Figure 7-3 SparkFun Sound Detector

## Audio Nano

I used an Arduino Nano that I purchased on Amazon to control the LEDS and to process the Gate input from the SparkFun sound detector. When the SparkFun sound detector gate is active, I illuminate the mouth LED. When the SparkFun sound detector gate is inactive, I extinguish the mouth LED.

The Arduino Nano also controls the Chest light LED and ramps this up and down. The eye LEDS are also controlled by the Arduino Nano and will blink the eyes at a random interval.

**<u>Power</u>**

The remote-control box requires between (9 VDC and 24 VDC) to operate and the Audio Playback device requires between (8 VDC – 24 VDC) to operate. The Arduino and LEDS require 5 VDC to operate. To resolve this easily, I used a (BINZET DC Converter Step Down Regulator 5V Regulated Power Supplies Transformer Converter (5V 3A 15W)). This device converts the 12 VDC input to 5 VDC. By using this device, I have both of the required voltages available for use.

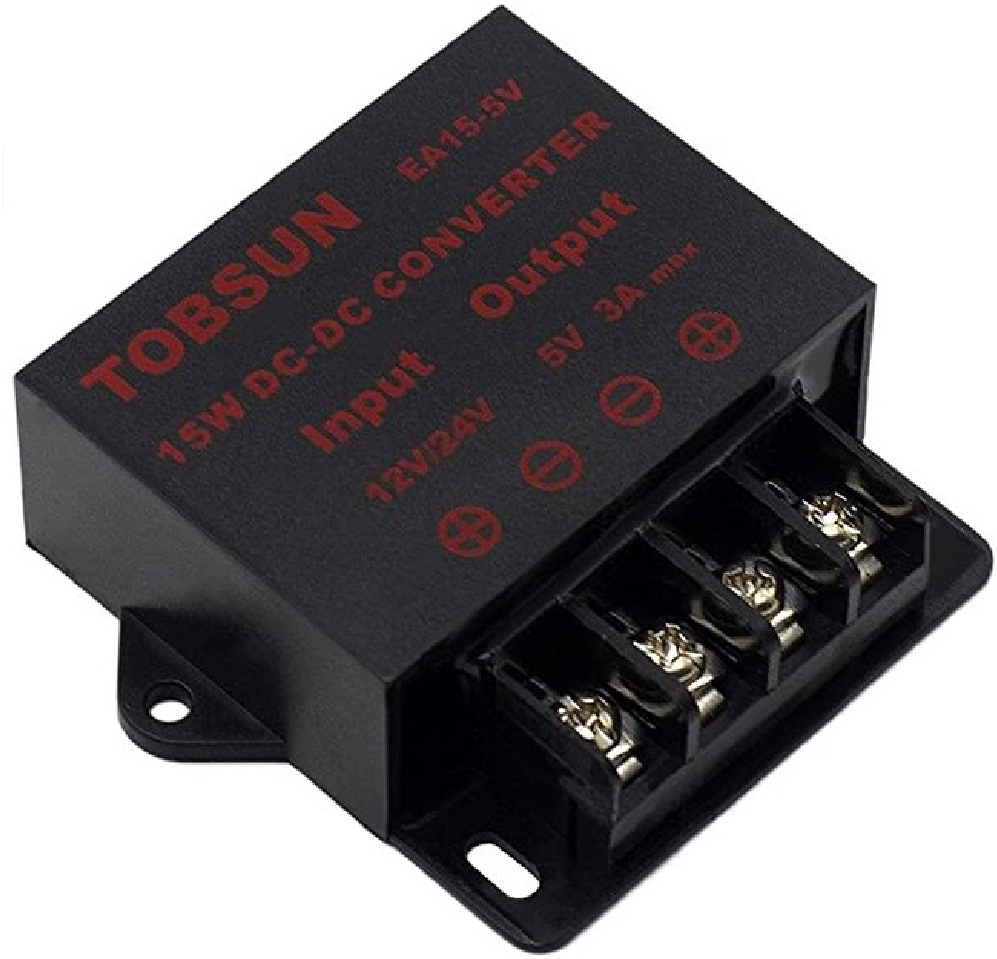
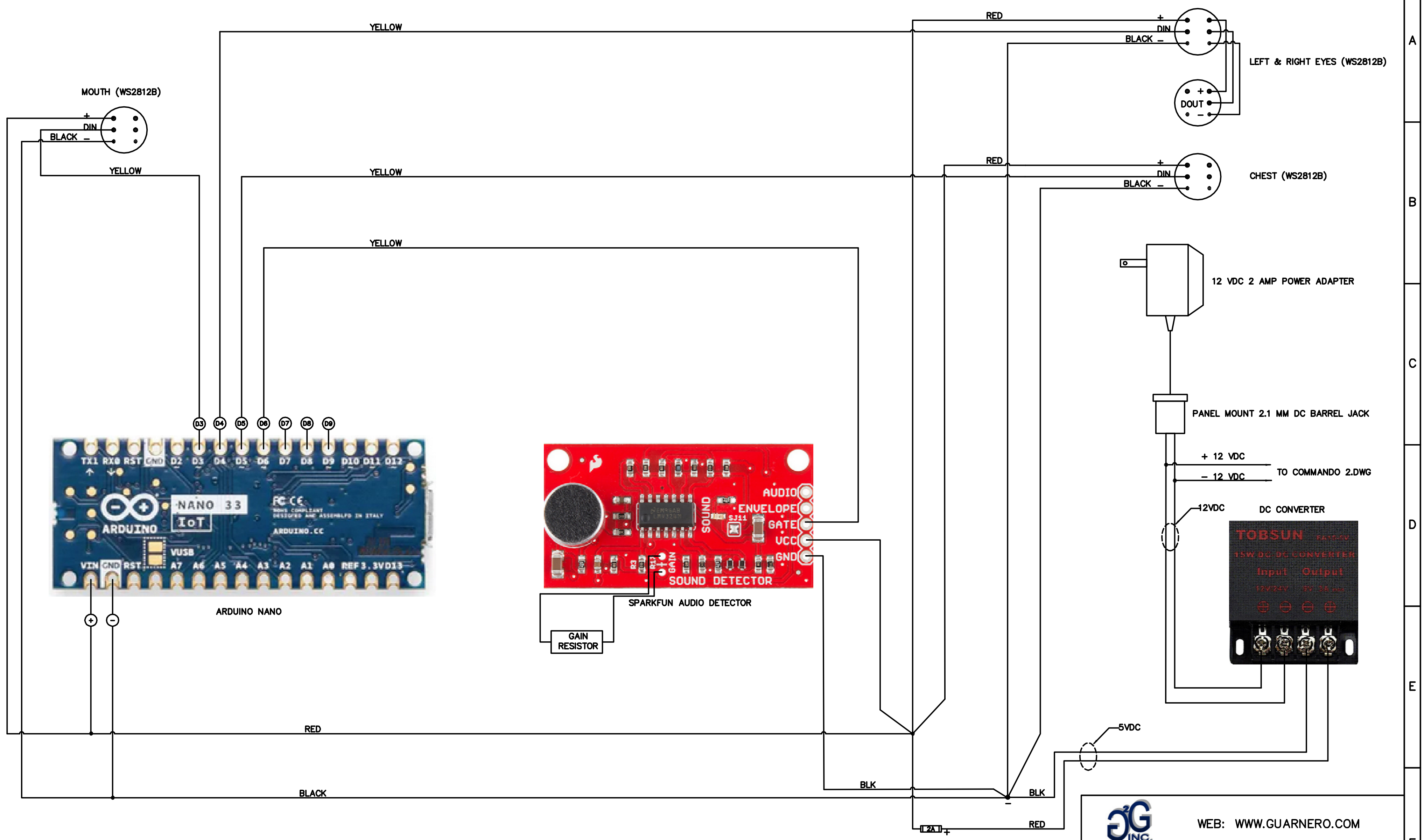


Figure 7-4 Step Down Regulator



TOLERANCES

DO NOT SCALE DRAWING

±1/16"
±1/16"
±0.010"
±1/60°

| REV | DATE | DESCRIPTION | BY | CHKD BY | ITEM NO. |
|---|---|---|---|---|---|

WEB: WWW.GUARNERO.COM

COMMANDO DROID VOICE AND WS2812B WIRING

COMMANDO

COMMANDO 1.DWG

JOHN GUARNERO

JOHN GUARNERO

JOHN GUARNERO

JOHN GUARNERO

MAY 22, 2020

JOHN GUARNERO

AS-BUILT

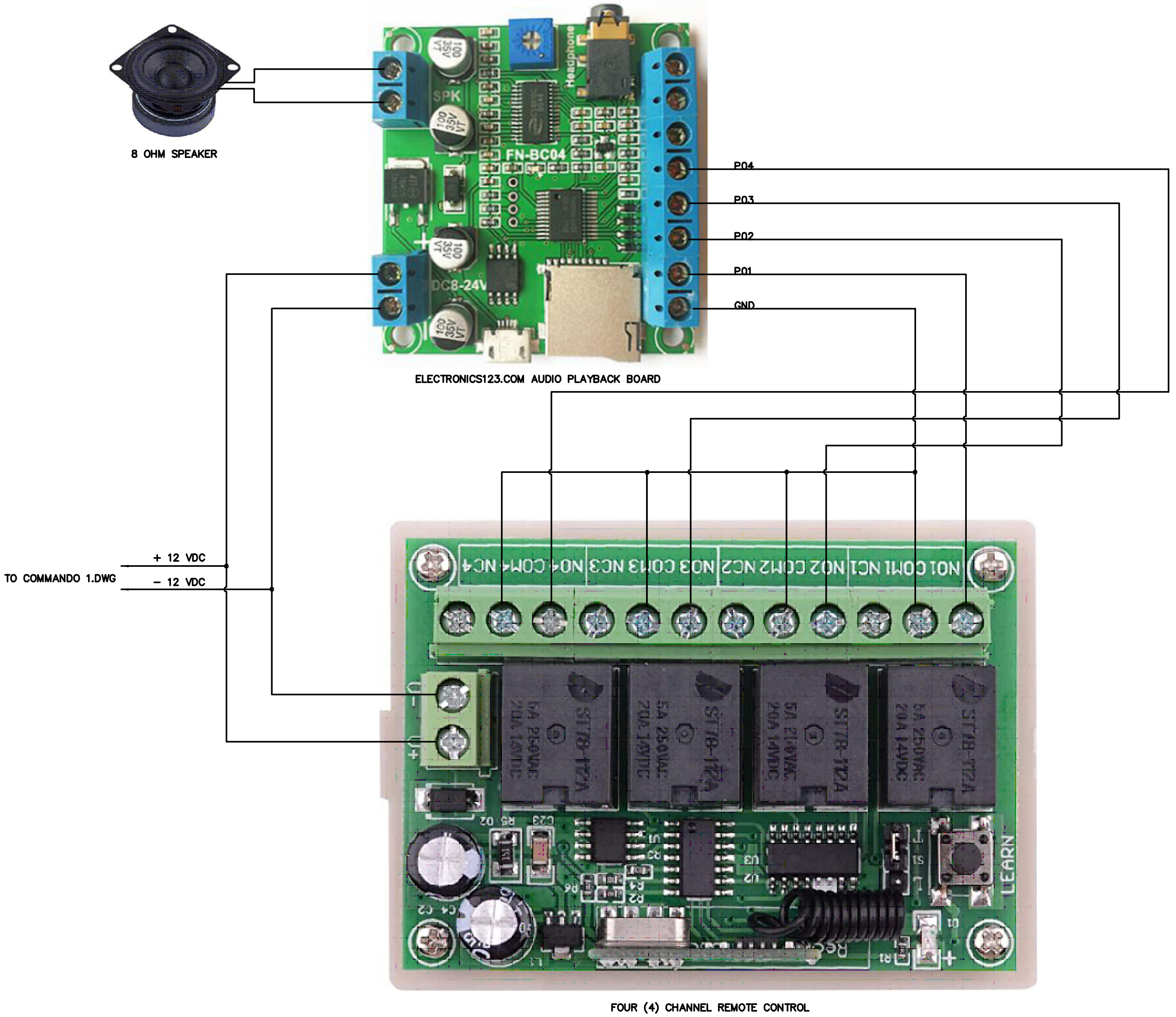
D

COMMANDO

SHEET 1 OF 2

REVISION

A


8 OHM SPEAKER
SPK
FN-BC04
Headphone
DC8-24V
P04
P03
P02
P01
GND
ELECTRONICS123.COM AUDIO PLAYBACK BOARD
TO COMMANDO 1.DWG
+ 12 VDC
− 12 VDC
NC4 COM4 NO4 NC3 COM3 NO3 NC2 COM2 NO2 NC1 COM1 NO1
+
−
LEARN
FOUR (4) CHANNEL REMOTE CONTROL

TOLERANCES
DO NOT SCALE DRAWING
±1/16"
±1/16"
±0.010"
±1/60'

| REV | DATE | DESCRIPTION | BY | CHKD BY | ITEM NO. |
|---|---|---|---|---|---|
| | | | | | |

WEB: WWW.GUARNERO.COM

COMMANDO DROID VOICE AND WS2812B WIRING

COMMANDO

COMMANDO 2.DWG

JOHN GUARNERO
JOHN GUARNERO
JOHN GUARNERO
JOHN GUARNERO
JOHN GUARNERO
MAY 22, 2020
AS-BUILT

D
COMMANDO

SHEET 2 OF 2

REVISION A