# Talking with Random Head Movement Skull Instructions

First let me thank you for attempting this project. I started this project 3 years ago and back then I knew nothing about programing Arduino or C code or anything. I still know very little and this was the vehicle I used to learn to program micro controllers.

I made several iterations of this and ultimately decided to go with a hybrid of a design by Mike North. His original design is available on his site buttonbanger.com under the tab, Jawduino. In order to complete this project, you are going to need some inexpensive modules and some basic soldering skills/equipment. We will be "hacking" one of the modules for this project.

Supplies:
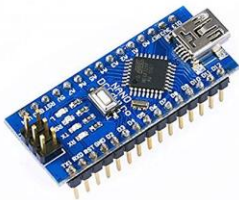
Arduino Nano or Uno (I prefer Nano's because they are small and fit inside props easily)

DF mini MP3 player (sometimes called TF-16P) these are cheap, work good and the code is specific to this player.

Also a mini SD card with audio files that you want the jaw servo to "talk" to. These files need to be named properly as well. The file names should be 001, 002,003 etc. as the player only recognizes numerical names in this format with the preceding zeros to 3 decimal places. You can have as many as 100 files if you want.
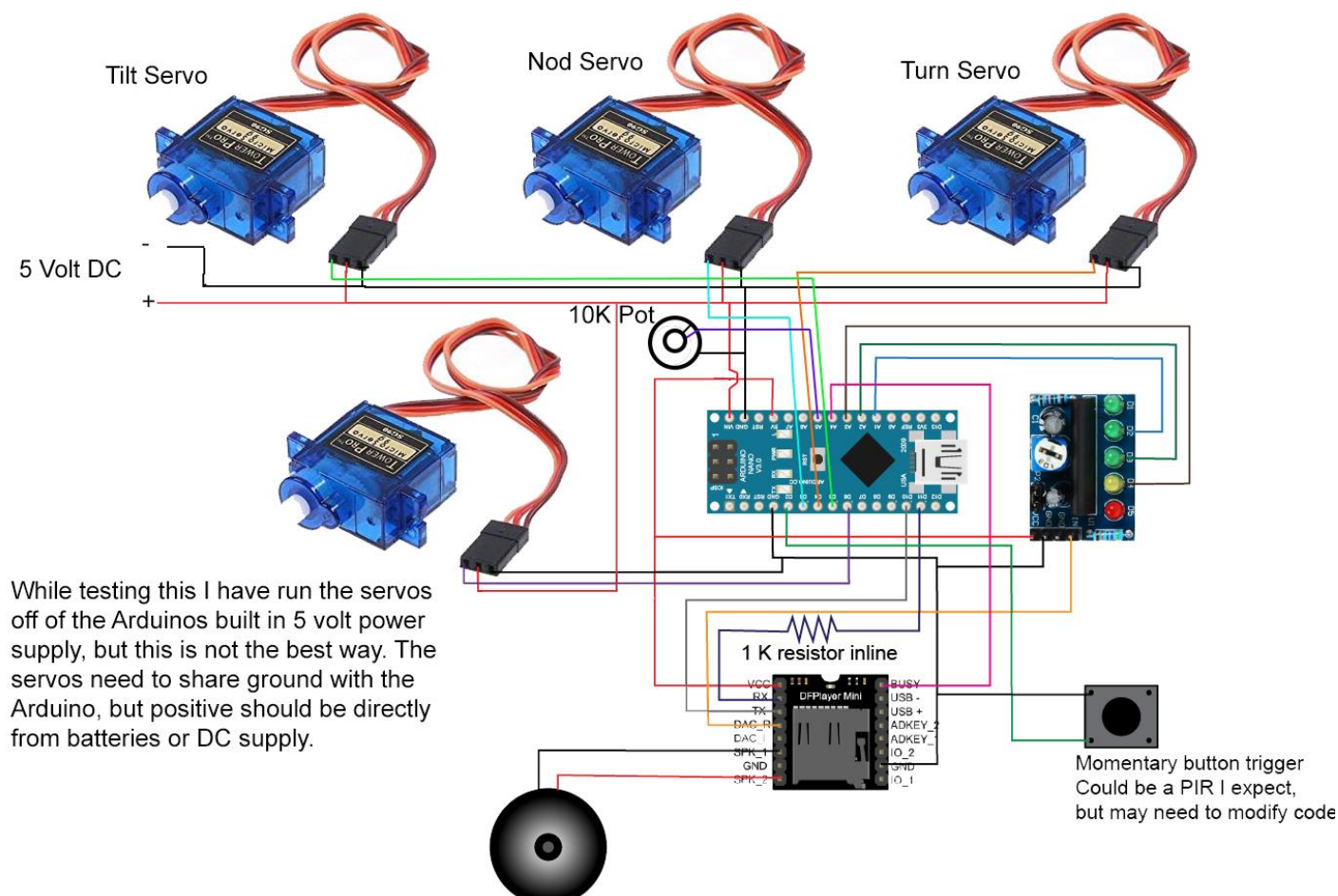
A KA2284 Module this is a small signal meter for audio devices and is the module we are going to hack to get our signal for the jaw to respond to.

Also, we will need a speaker, hookup wires and a button or PIR motion detector and a 10K Potentiometer.

I have not yet tried the PIR in the circuit, but it should work. You may need to invert the signal from it. Some PIR's have a jumper for this purpose. We are using a grounding of pin 2 to signal the start of a cycle.

Here is the layout:

Tilt Servo

Nod Servo

Turn Servo

5 Volt DC

−

+

10K Pot

1 K resistor inline

DFPlayer Mini

VCC
RX
TX
DAC_R
DAC_L
SPK_1
GND
SPK_2

BUSY
USB -
USB +
ADKEY_1
ADKEY_2
IO_2
GND
IO_1

While testing this I have run the servos off of the Arduinos built in 5 volt power supply, but this is not the best way. The servos need to share ground with the Arduino, but positive should be directly from batteries or DC supply.

Momentary button trigger
Could be a PIR I expect, but may need to modify code

You can power the Arduino with 5 to 12 Volts DC supplied to the Arduino Vin and Ground pins. But the servos need to be limited to 5 Volts. The 1 K resistor in-line with RX is necessary to reduce major audio noise over the speaker.

Servo Limit Settings

The first thing to do is load the Servo limit code onto the Arduino. I recommend using the IDE serial window for feedback as to what part of the servo limit setup you are in. Hardware wise all you need to hook up for this part is the momentary button and the 10K potentiometer and the three random movement servos (Nod, Turn, Tilt). The 10K pot is used to adjust each servo setting to the physical spot you want it. This is achieved by turning the pot until the skull moves to the desired position. One servo is setup at a time. Each servo has three settings and, in order, they are servo Lower Limit, Servo Center servo, Upper Limit. Each servo will step through these three settings in order from Nod servo, to Turn servo, to Tilt servo. Then the code advances to an option to store (write to EEprom) the set values or skip and go back to make the adjustments again. Once you are happy with the servo position settings then write them and the Serial window will say "Safe to exit Setup." You can then unplug the USB cable and or load the second sketch which is: 3AxisSkullComplete.ino which just runs the skull and plays the audio.

Jaw Movement
Once the whole thing is working you have to tune the signal meter module using the onboard potentiometer to get the signal just right for the jaw movement. If you just fiddle with it while the audio is going you will get

it looking good. I might add that the signal to the servo is meant to open the jaw, not close it. Meaning, the hardware of the jaw should be normally closed and opens when the audio starts getting louder on the hacked signal meter.

RX from the MP3 module has a 1K Resistor inline and connects to D11 on the Arduino. The 1 K resistor in-line with RX is necessary to reduce major audio noise over the speaker. The rest of the connections are as follows: MP3 Tx = Arduino D10, MP3 Busy = Arduino A4, MP3 DAC-R = Input pin on KA2284 Module, Momentary Trigger button/PIR = Arduino D2, Servo Signal = D6. Everything needs a common Ground connection, so run a ground wire between the following: MP3 player, Switch/PIR, KA2284, Arduino, Servo. For Power the Arduino Has the ability to deliver 5V @ 1Amp. This is enough for small servos, but not big ones. Best practice is to power the servo and Arduino from a separate power supply. I recommend a DC-DC Buck converter module as they are inexpensive and will output 3 to 5 Amps depending on the model. Make sure to tune the Buck convertor to 5V output before connecting it up! To use one, you would just connect the ground to the previously described ground loop and Positive to the servo red wire (which is always the center of the 3) and to Arduino Vin. To power the other modules run a power loop from Arduino 5V to MP3 VCC and KA2284 VCC.

The speaker is small (less than 3 Watt) output.

The hack:
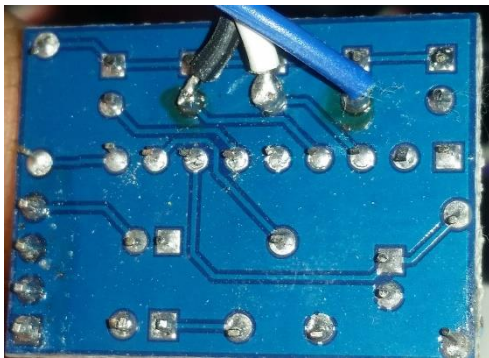
What you need to do to the KA2284 module is tie into the signal that goes to the 3 middle LED's. If you look on the back of the module (you may need a magnifying glass) you can see the circuit board traces coming from the IC chip to the LED's. These LED's connect to the IC and ground on their leads. You want to connect to the IC/LED side of the LED, NOT the LED/Ground side. You could even un solder the LED's and remove them which would leave you with a nice through hole to solder the wires to, but you do run the risk of damaging the board more when you unsolder components. You decide. The bottom line is to get the output of the three channels from IC chip sent into the Arduino for signal processing. The 3 wires from this hack need to get to the correct pins or this won't work. On the front the LED's are labeled D2, D3, and D4. They need to end up hooked up as follows: D2 connected to A1 on the Arduino, D3 connected to A2 on the Arduino, D4 connected to A3 on the Arduino.

The sensitivity of the movement is controlled by the Potentiometer located on the KA2284 board. You will have to fiddle with it based on the loudness of your MP3s.

That should be it, if you have problems or issues please let me know and I will do my best to help.

copyright 2022 Kenny Chapman

Batbuddy on hauntforum.com   MrBatbuddy on YouTube