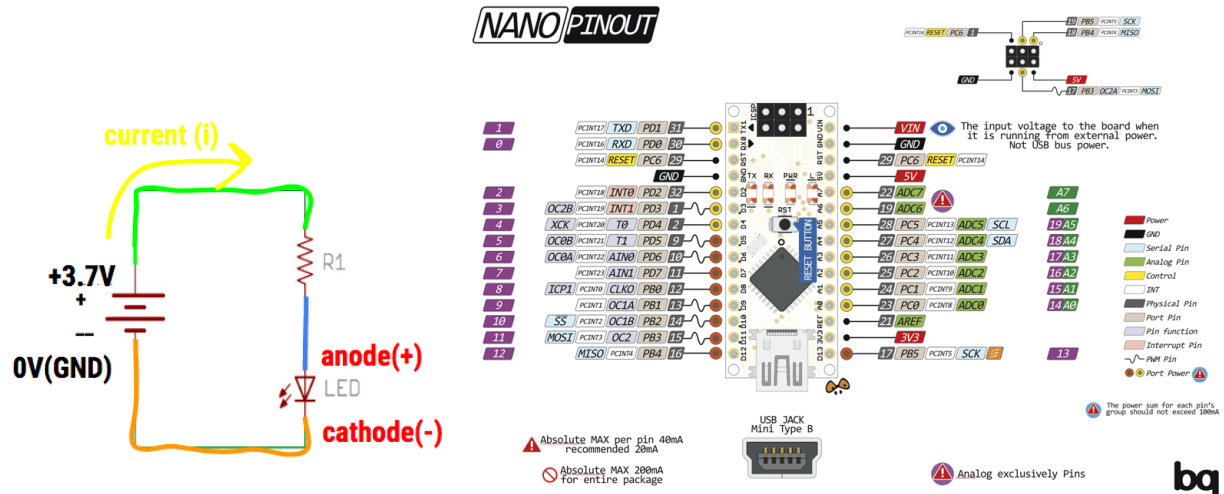
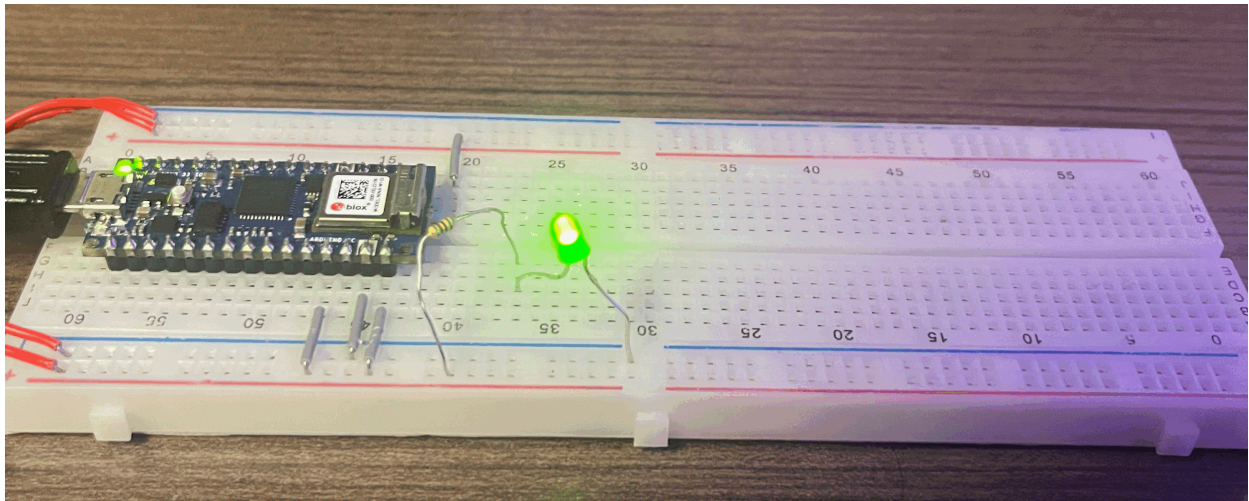


Ipoduino Log Book

Goal 1: Use Arduino Nano USB power supply to light up a led



Implementation:



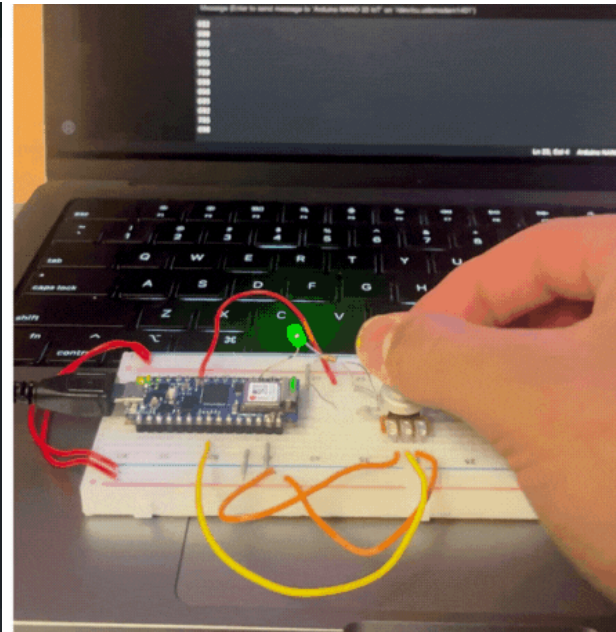
Take Away: Using the PinOUT diagram, only the VIN and 3V3 pins were able to power the LED without relying on instructions from the microcontroller.

Goal 2: Use a Potentiometer to control the state of a LED

Implementation:

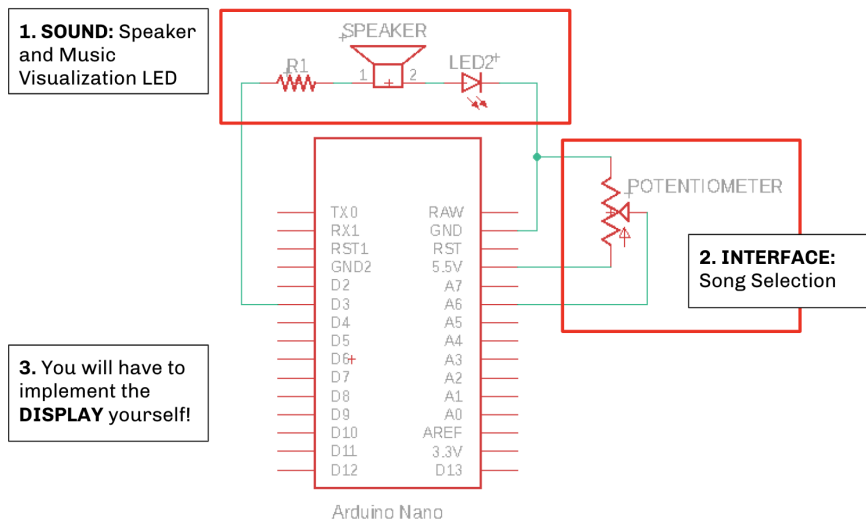
Details: Code is written to read the value of the potentiometer, print the value in the Serial Monitor and keep the LED on (HIGH) if the value is greater than 300, and off (LOW) if the value is less than or equal to 300.

```
potLED.ino
1  #define POT 18
2  #define LED 7
3
4  void setup() {
5    // put your setup code here, to run once:
6    pinMode(POT, INPUT);
7    pinMode(LED, OUTPUT);
8    Serial.begin(9600);
9
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14   Serial.println(analogRead(POT));
15
16
17   if(analogRead(POT)>300)
18     digitalWrite(LED,HIGH);
19   else{
20     digitalWrite(LED,LOW);
21     delay(1000);
22   }
23 }
24
25
```



Goal 3: Make speaker circuit, play 2 different songs on the speakers, and have a display to indicate the current song

Speaker Circuit



I first made this circuit and wrote up the code to play Despacito and used the potentiometer to select between playing the song at 2 different tempos.

The code relies on a pitches.h file to translate the different notes that appear in sheet music to the constant values that produce the correct sound.

Then I created a Despacito() function to arrange the notes according to the sheet music and used a Play() function to play each note for a specified time.

```

void play(int note, int dur) {
  tone(SPEAKER, note);
  delay(dur * NOTE_DUR);
  noTone(SPEAKER);
  delay(dur * NOTE_DUR / 3);
}

void Despacito(int Note_dur) {
  // Set tempo to be faster
  NOTE_DUR = Note_dur;
  play(NOTE_D4, 1);
  play(NOTE_FS4, 1);
  play(NOTE_B4, 1);
  play(NOTE_D5, 1);
  play(NOTE_CS5, 1);
  play(NOTE_B4, 1);
  play(NOTE_AS4, 1);
  play(NOTE_B4, 16); //m2
  delay(NOTE_DUR*6);
  play(NOTE_B4, 6);
  play(NOTE_CS5, 6);
  play(NOTE_D5, 6);
  play(NOTE_E5, 6); //m3
  play(NOTE_FS5, 7);
  play(NOTE_D5, 7);
  play(NOTE_FS5, 8);
  play(NOTE_D5, 8);
  play(NOTE_FS5, 22);
  delay(NOTE_DUR*4);
  play(NOTE_FS4, 2); //m4
  play(NOTE_B4, 4); //m5
  play(NOTE_D4, 2);
  play(NOTE_FS4, 4);
  play(NOTE_B4, 2);
  play(NOTE_D4, 4);
  play(NOTE_B4, 4);
  play(NOTE_D4, 2);
  play(NOTE_AS4, 2);
  play(NOTE_D5, 2);
  play(NOTE_CS5, 2);
  play(NOTE_B4, 2);
  play(NOTE_A4, 2);
  play(NOTE_B4, 4); //m6
  play(NOTE_D4, 2);
  play(NOTE_G4, 4);
  play(NOTE_B4, 2);
  play(NOTE_D4, 4);
  play(NOTE_B4, 4);
  play(NOTE_D4, 2);
  ... rest of song
}

```

Considering the LCD pinout diagram,

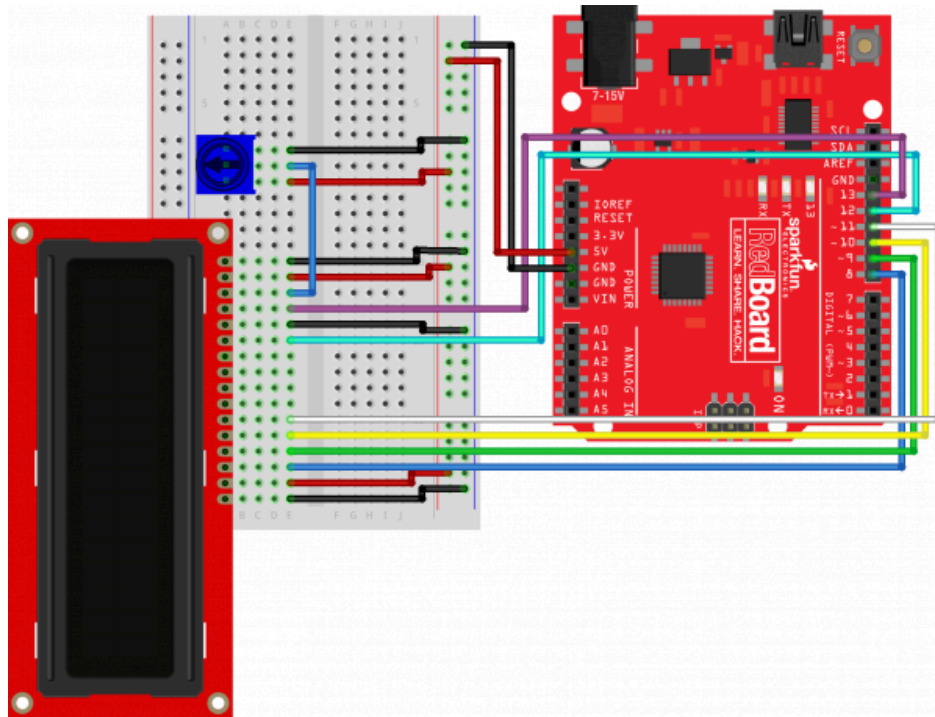


Pin Descriptions:
Pin 1 -- Ground
Pin 2 -- VDD Power for the LCD
Pin 3 -- Contrast Adjust
Pin 4 -- Register Select (RS)
Pin 5 -- Read / Write Select (R/W)
Pin 6 -- Enable

Pins 7-10 -- data lines d0 - d3 (not used)
Pins 11 - 14 -- data lines d4 - d7 (data transferred in 4-bits at a time)*

Pin 15 -- Backlight Power
Pin 16 -- Backlight GND (GND)

I implemented the LCD circuit as advised in this sparkfun hook-up guide



In my case I used an Arduino NANO again instead of the Sparkfun RedBoard; however, I did still use the same for the data ports. The only major modification I made was routing the 3rd pin to a analog port of the NANO instead of to a potentiometer and using the analogWrite() function to set the Contrast Adjust directly, with less circuitry.

I then used the LCD to print out the current playing song and the time elapsed since last restart.

```

#include <LiquidCrystal.h>

//initialize the library by associating any needed LCD interface pin
//with the arduino pin number it is connected to
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

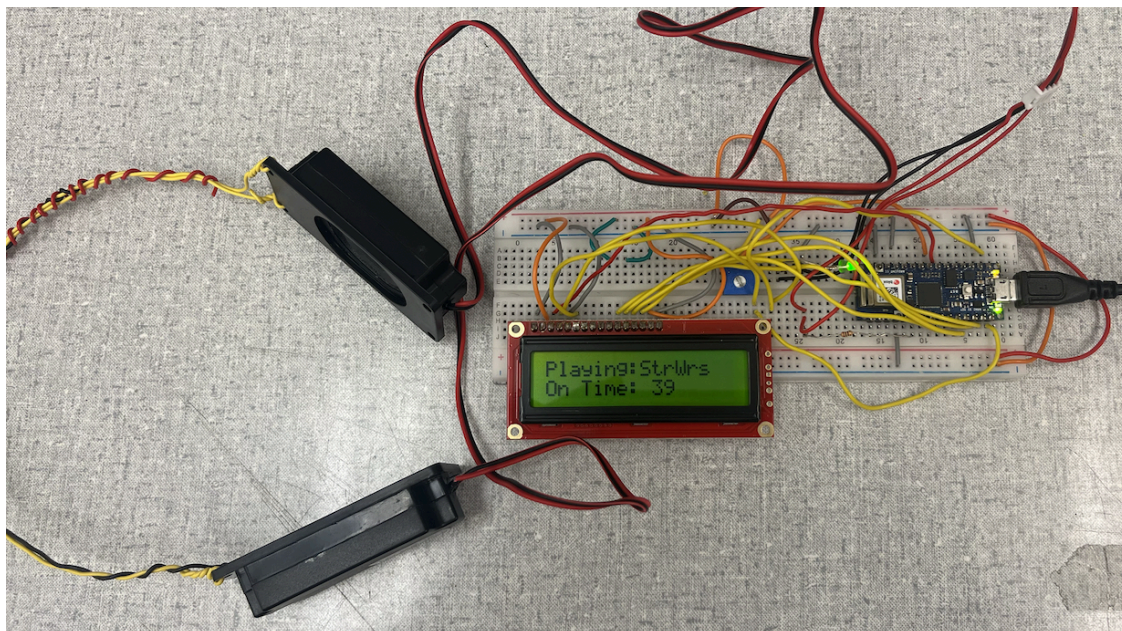
lcd.begin(16, 2);
lcd.print("Playing:");

lcd.setCursor(8, 0);
if (analogRead(POT)>400){
  lcd.print("StrWrs");
  StarWars(140);
}
else{
  lcd.print("Dspcto");
  Despacito(70);
}

```

Finally, I created a StarWars() function to play Cantina Band from Star Wars using similar techniques as the Despacito function. And I used the potentiometer input as a condition in an If control flow to select between playing Despacitio and StarWars.

Final Circuit and Demonstration:



[Ipoduino_Demo.MOV](#)

