Singing Christmas Lights with Heat and Remote Activation

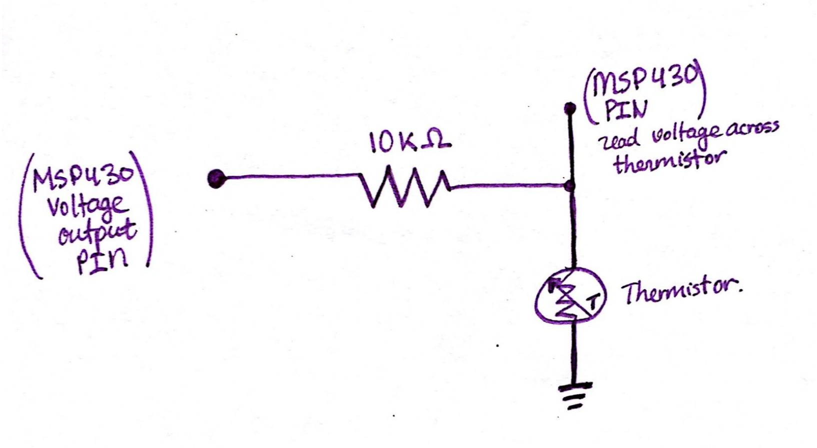
Enkhtushig Namkhai and Michelle Chang

Overview

This project, which is unified around the theme of Christmas, features four key parts that allow the user to interact with its components. By touching a thermistor, users are able to activate a set of red LEDs and invoke a volume-controlled sound system that plays “We Wish You a Merry Christmas.” In the case that users are too lazy to walk over to the display, our device also provides an interactive mobile feature that allows one to turn on the display by sending a signal from a mobile device to the msp430 microcontroller via Bluetooth. It is our hope that this project will enable users to further enhance the joy of Christmas within their own homes.

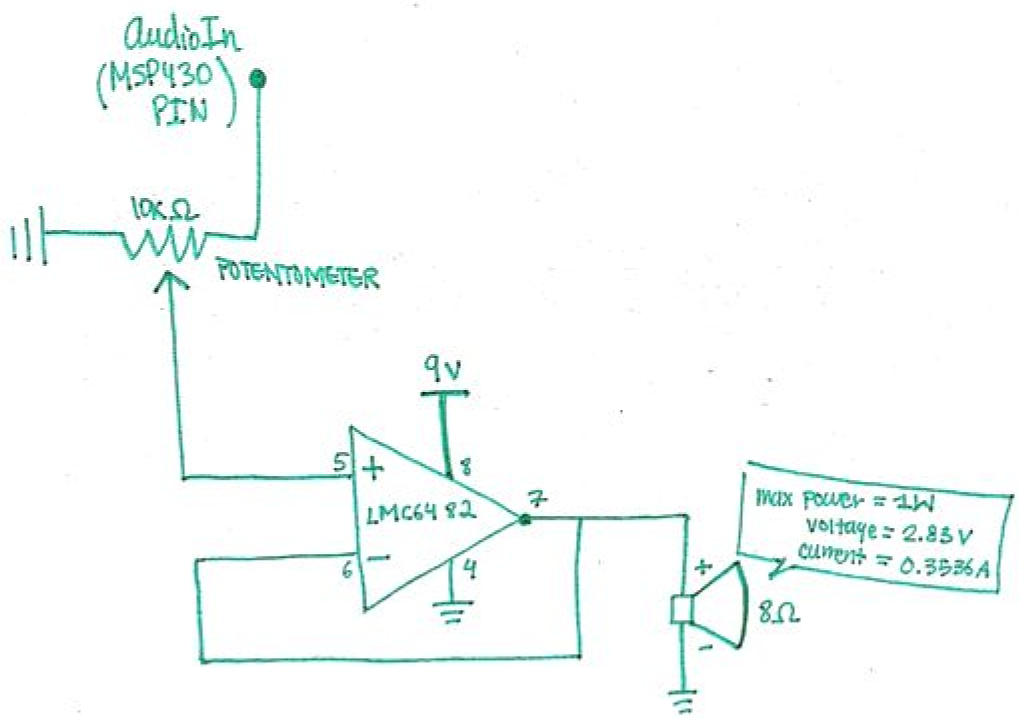
Features Provided

Heat-Activated System (Key Component: Thermistor):



Users can activate our project by touching a thermistor. We calculate the temperature of the user by measuring the voltage through a voltage divider, in which the thermistor is connected to a 10 kΩ resistor. The msp430 reads this voltage value, which is then converted, through a series of calculations, into Celsius temperature. Light and sound magic is only activated if the Celsius temperature is over 30 degrees.

Volume Control (Key Component: Potentiometer):

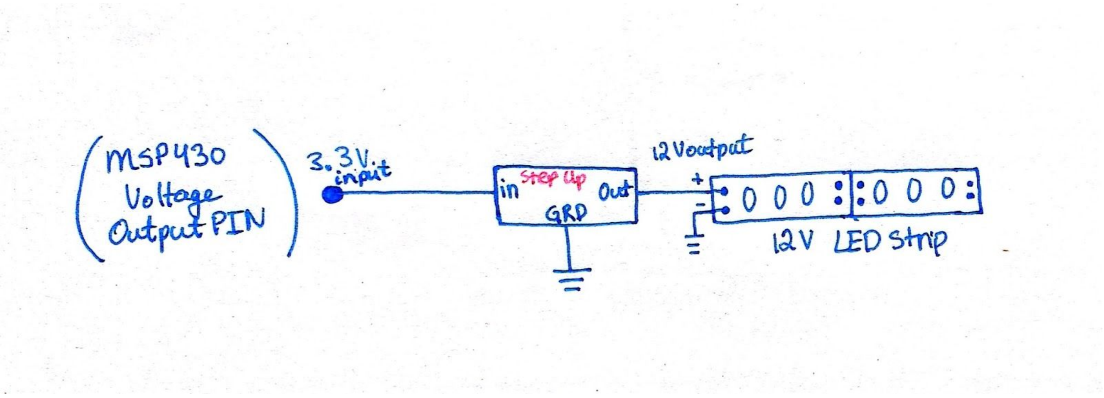


For this feature, we took the basic sound circuit employed in lab and inserted a potentiometer in between the msp430 and the LM386 op-amp. The potentiometer varies the resistance encountered by the signal sent from the msp430 to the op-amp. In doing so, it allows users to control the volume of sound emitted by the speaker. Additionally, to increase the overall sound (so volume control would have a greater effect), we limit the op-amp to 9 V; this is a valid action as we took precautions to measure the op-amp output voltage and discovered that its values never exceed 2.83 V (around the maximum voltage the speaker can handle). We found the maximum voltage the speaker can withstand using its power limit of 1W and resistance of 8Ω.

Music (Key Component: Software):

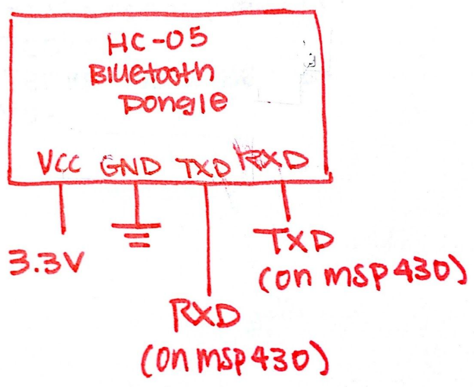
Our robot plays a Christmas song when input (thermistor or remote activation) is applied. We used the music notes built into the pitches.h library (provided by Arduino open source) to hardcode “We Wish You a Merry Christmas.” To control note duration, we specify each note type as a portion of 1 second (1000 ms). For example, quarter notes are ¼ of a second and eighth notes are ⅛ of a second.

Lights (Key Components: 12V Step-Up & Red LED Strip):



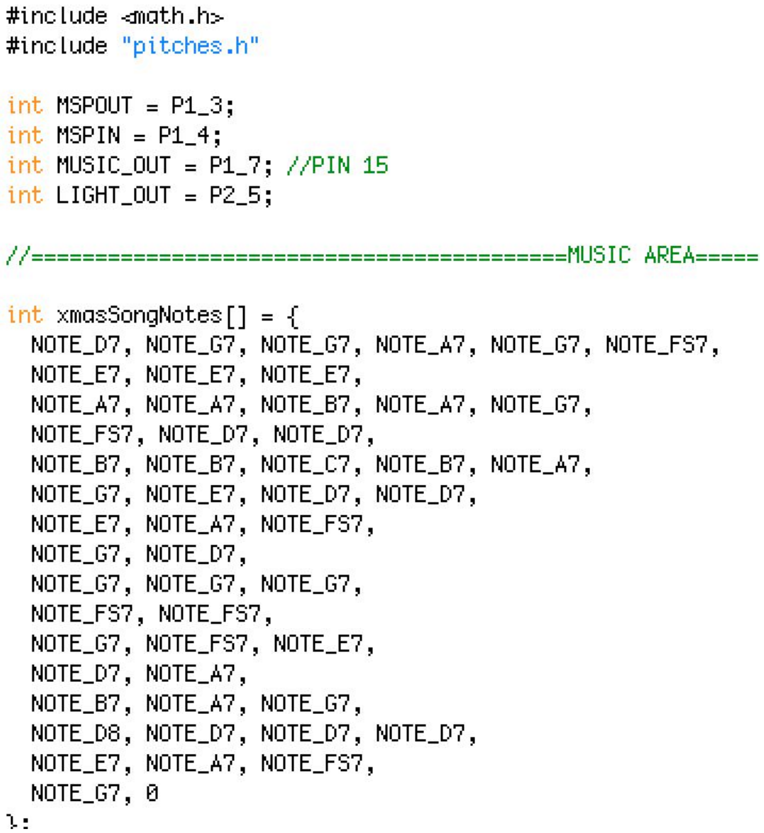
When the circuit receives the input (Thermistor, Remote), the Output pin to the Step Up is turned on, by writing HIGH to the pin. Then pin then sends 3.3 Volts to the input of the Step-Up and the Step-Up, increases the Voltage to be 12 Volts so that the 12 Volt LED Strip is able to turn on once someone touches the Thermistor or once someone activates it by bluetooth.

Bluetooth Remote Activation (Key Component: HC-05 Wireless Bluetooth Host Serial Transceiver Module):



Our robot also supports activation by bluetooth connection. When our user is feeling lazy or if our user does not have a high enough body temperature to turn on the system via heat, they can send a signal (“L, M, or A”) from their phone and turn the lights on for one minute (with “L”), play only the music (with “M”), or activate both (with “A”) from afar. We used the HC-05 device to connect the MSP430 to our phone by bluetooth. Remote activation is provided by the Ardudroid Android app.

Code

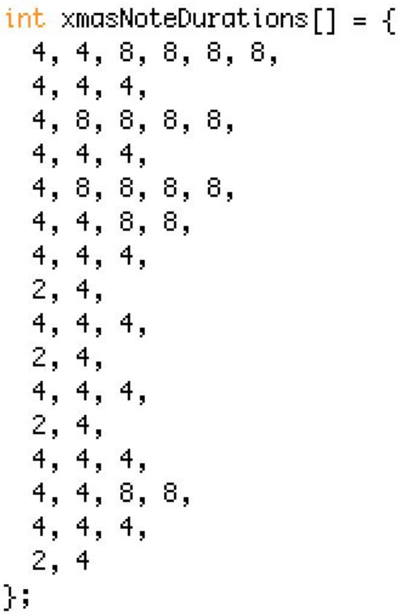


```
#include <math.h>
#include "pitches.h"

int MSPOUT = P1_3;
int MSPIN = P1_4;
int MUSIC_OUT = P1_7; //PIN 15
int LIGHT_OUT = P2_5;

//==========================================MUSIC AREA=====

int xmasSongNotes[] = {
  NOTE_D7, NOTE_G7, NOTE_G7, NOTE_A7, NOTE_G7, NOTE_FS7,
  NOTE_E7, NOTE_E7, NOTE_E7,
  NOTE_A7, NOTE_A7, NOTE_B7, NOTE_A7, NOTE_G7,
  NOTE_FS7, NOTE_D7, NOTE_D7,
  NOTE_B7, NOTE_B7, NOTE_C7, NOTE_B7, NOTE_A7,
  NOTE_G7, NOTE_E7, NOTE_D7, NOTE_D7,
  NOTE_E7, NOTE_A7, NOTE_FS7,
  NOTE_G7, NOTE_D7,
  NOTE_G7, NOTE_G7, NOTE_G7,
  NOTE_FS7, NOTE_FS7,
  NOTE_G7, NOTE_FS7, NOTE_E7,
  NOTE_D7, NOTE_A7,
  NOTE_B7, NOTE_A7, NOTE_G7,
  NOTE_D8, NOTE_D7, NOTE_D7, NOTE_D7,
  NOTE_E7, NOTE_A7, NOTE_FS7,
  NOTE_G7, 0
};
```



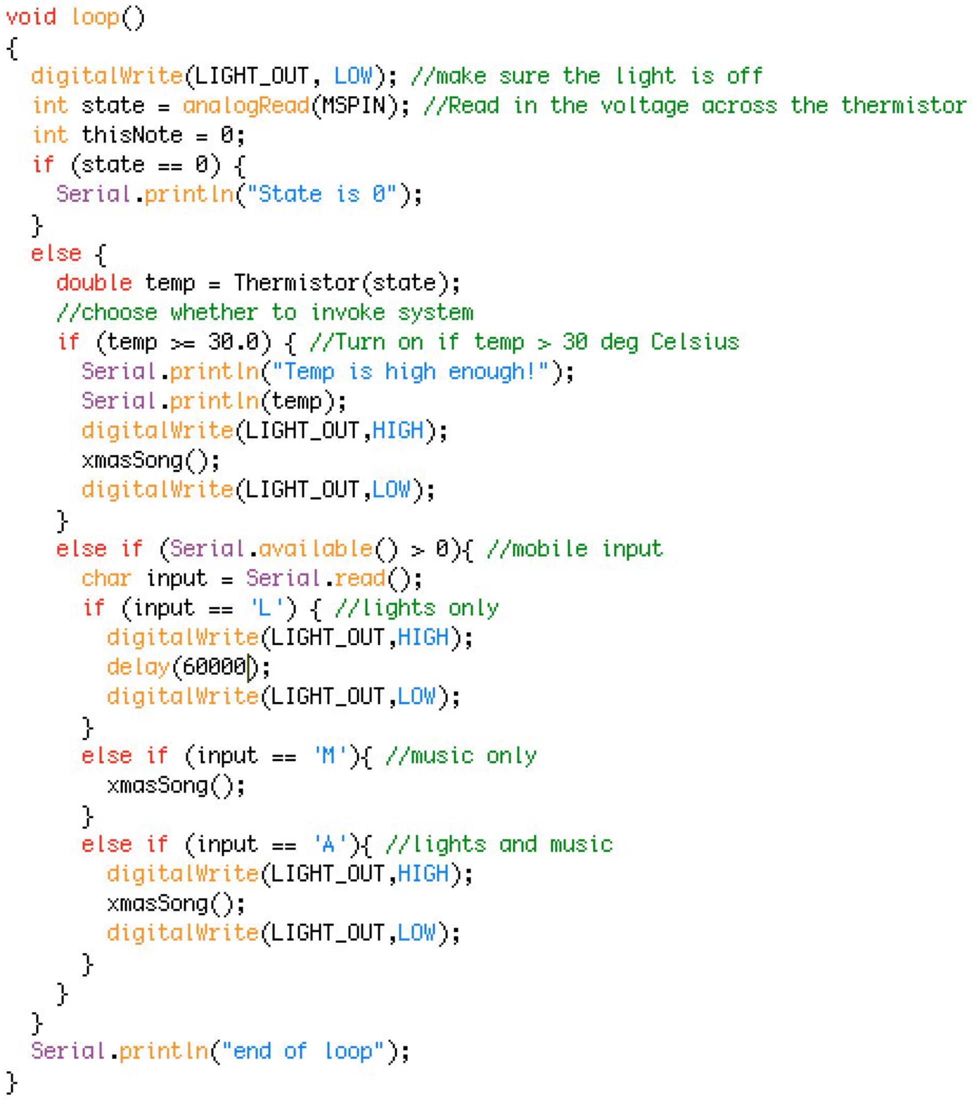
```
int xmasNoteDurations[] = {
  4, 4, 8, 8, 8, 8,
  4, 4, 4,
  4, 8, 8, 8, 8,
  4, 4, 4,
  4, 8, 8, 8, 8,
  4, 4, 8, 8,
  4, 4, 4,
  2, 4,
  4, 4, 4,
  2, 4,
  4, 4, 4,
  2, 4,
  4, 4, 4,
  4, 4, 8, 8,
  4, 4, 4,
  2, 4
};
```

```
void xmasSong() {
    // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < sizeof(xmasSongNotes)-1; thisNote++) {
    // to calculate the note duration, take one second
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000/xmasNoteDurations[thisNote];
    tone(MUSIC_OUT, xmasSongNotes[thisNote],noteDuration);
    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(MUSIC_OUT);
  }
  return;
}

//=========================================END MUSIC AREA=======================

/* Calculates and returns the temperature of the thermistor.
 * Uses the simplified Steinhart-Hart Thermistor Equation:
 * 1 / Kelvin_temp = (1/B)*ln(R/R_0) + (1/T_0)
 * B = constant specific to thermistor model
 * R = resistance of thermistor
 * R_0 = thermistor resistance at room temperature
 * T_0 = room temperature = 25 C = 288.15 K
*/
float Thermistor(int RawADC) {
 float Temp;
 float B = 3600;
 float r_in = 10000.0*(1/((1024.0/RawADC)-1));
 Temp = logf(r_in/10000.0); //Equivalent of ln(R/R_0), R = resistance of therm
 Temp = 1/((1/B)*Temp + (1/298.15)); //T_0 = 25 C = 298.15 K
 Temp = Temp - 273.15; // Convert Kelvin to Celcius
 return Temp;
}

void setup()
{
  Serial.begin(9600);
  pinMode(MSPOUT, OUTPUT); //msp430 sends signal to circuit through here
  pinMode(MSPIN, INPUT); //msp430 receives signal from circuit here
  pinMode(LIGHT_OUT, OUTPUT); //msp430 turns on lights here
}
```



```
void loop()
{
  digitalWrite(LIGHT_OUT, LOW); //make sure the light is off
  int state = analogRead(MSPIN); //Read in the voltage across the thermistor
  int thisNote = 0;
  if (state == 0) {
    Serial.println("State is 0");
  }
  else {
    double temp = Thermistor(state);
    //choose whether to invoke system
    if (temp >= 30.0) { //Turn on if temp > 30 deg Celsius
      Serial.println("Temp is high enough!");
      Serial.println(temp);
      digitalWrite(LIGHT_OUT,HIGH);
      xmasSong();
      digitalWrite(LIGHT_OUT,LOW);
    }
    else if (Serial.available() > 0){ //mobile input
      char input = Serial.read();
      if (input == 'L') { //lights only
        digitalWrite(LIGHT_OUT,HIGH);
        delay(60000);
        digitalWrite(LIGHT_OUT,LOW);
      }
      else if (input == 'M'){ //music only
        xmasSong();
      }
      else if (input == 'A'){ //lights and music
        digitalWrite(LIGHT_OUT,HIGH);
        xmasSong();
        digitalWrite(LIGHT_OUT,LOW);
      }
    }
  }
  Serial.println("end of loop");
}
```