Arduino Beehive Temperature and Humidity Monitors

ECE Honor Projects

By Kefan Tu

I. Introduction

1. Problem Description

The living conditions for bees are big concern for beekeepers, and temperature and humidity might be two factors that affect the beehive environment a lot. The project aims to figure out what exact values of temperature and humidity are best for the bees' living, according to regular changes in honey production.

2. Design Concept

In this project, a Grove Temperature and Humidity Sensor connected to the Grove Launchpad will be put inside the hive to measure the surrounding temperature and humidity, whose values will be shown on a Grove 4-digit-display. 4 load cells will be put under 4 corners of a hive to measure daily changes in the weight, from which we may indicate how the honey production is every day. Honey production is believed to be closely related to bees' health in this project. An operational amplifier LM358 will be used with the load cells to make the readings more obvious. After some conclusion is drawn from the measurement, I will determine which range of temperature and humidity is optimal for the bees. A buzzer is attached, which will make sound to get notice of beekeepers nearby when the temperature or humidity inside the hive is beyond the optimal range.

II. Analysis of Components

1. Characteristics of Sensors

The Grove Temperature and Humidity Sensor measures temperature from -40 to 80 degrees of Centigrade, and humidity from 5%-99%.

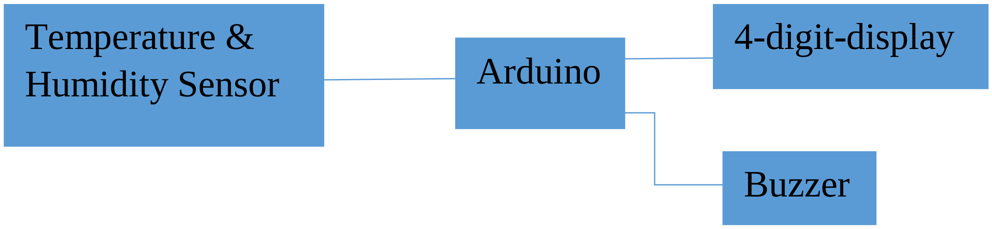
The capacity of load cell SEN-10245 is 50 kg; I hooked up the sensor to the Arduino so that I could measure output voltage by using the analog input on Arduino, when I put objects of known weight on the load cell. (I did not finish this step yet.)

2. Design Consideration

The output voltage from the load cells that we measure may changes obviously when we need to know about the honey production, so I hooked up an operational amplifier LM358N with it to amplifier the readings.

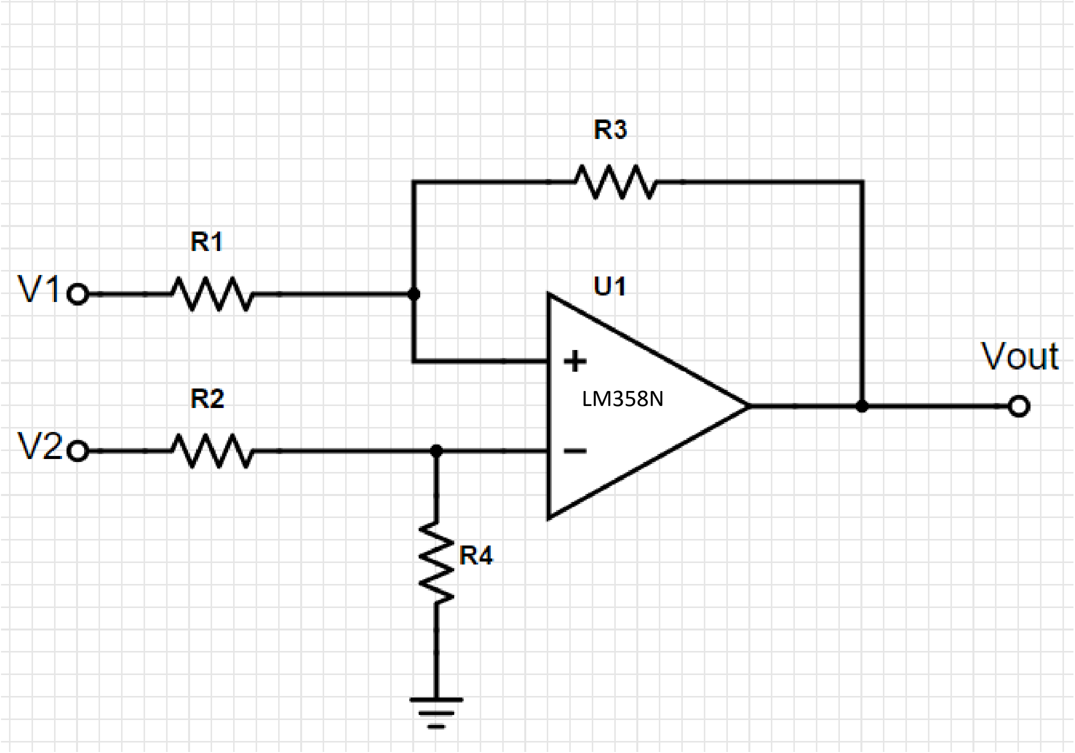
III. Design

1. Block Diagrams



The signal pin of temperature and humidity sensor is connected to pin 24; the 4-digit-display is attached to pin 38 and 39; the Grove Buzzer is connected to pin 36. The Grove LaunchPad receives feedbacks from the sensor and display them every 2 seconds on the 4-digit-display. The first two digits represent the temperature in degree Celsius, and the other two represent humidity in percent. The buzzer is set to function when temperature exceeds 30 degrees Celsius (30 may not be the value I finally find out; it is just for proving the design idea).

2. Circuit Schematics



The op-amp LM358N is power by +5V. V1 represents the output from the load cell and V2 is +5V. If R3=R4 and R1=R2, Vout=R4/R1*(V1-V2). Thus I choose 47 Ω for R1 and R2, and 4.7 kΩ for R3 and R4, so that the reading is amplified by 100.

3. Codes Used

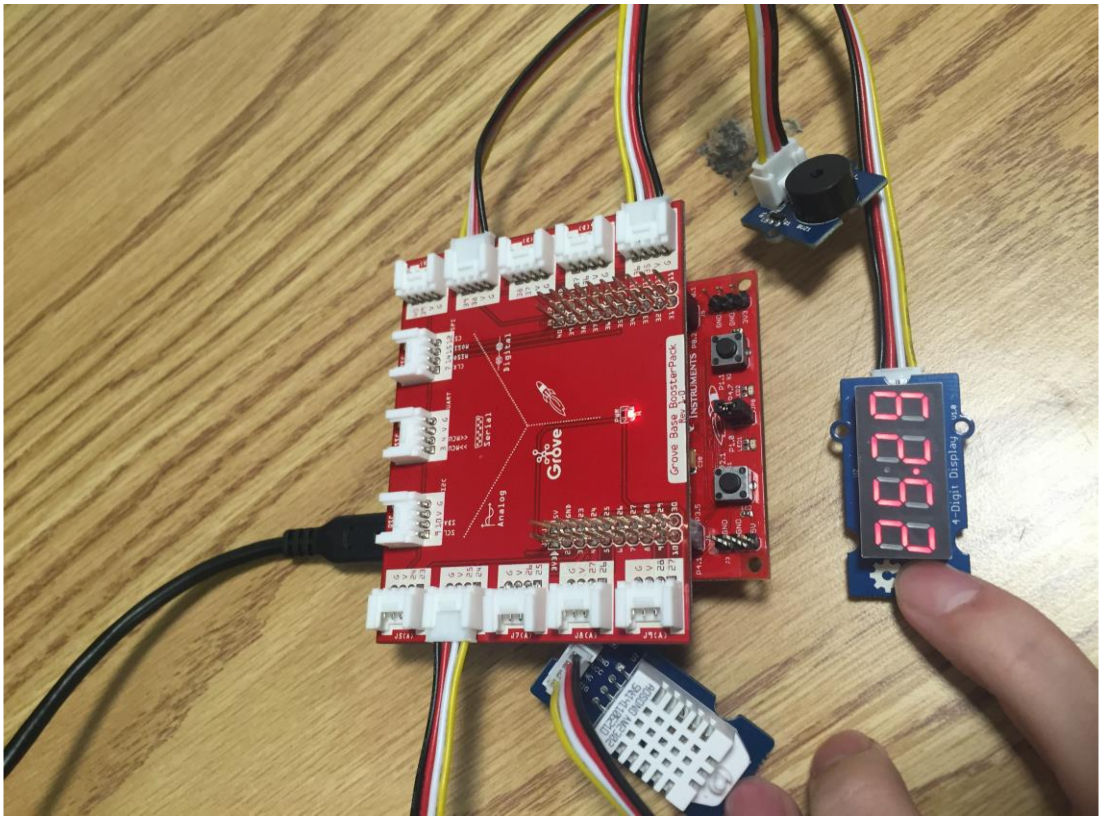
```
#include <DHT.h>
#include <TM1637.h>
#define CLK 39
#define DIO 38
#define TEMP_HUMI_PIN 24
TM1637 tm1637(CLK, DIO);
DHT dht(TEMP_HUMI_PIN, DHT22);
int8_t t_bits[2] = {0};
```

```
int8_t h_bits[2] = {0};
void setup()
{
    tm1637.init();
    tm1637.set(BRIGHT_TYPICAL);
    tm1637.point(POINT_ON);
    dht.begin();
    pinMode(RED_LED, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    int temperature = dht.readTemperature();
    int humidity = dht.readHumidity();
    memset(t_bits, 0, 2);
    memset(h_bits, 0, 2);
    t_bits[0] = temperature % 10;
    temperature /= 10;
    t_bits[1] = temperature % 10;
    h_bits[0] = humidity % 10;
    humidity /= 10;
    h_bits[1] = humidity % 10;
    tm1637.display(0, t_bits[1]);
    tm1637.display(1, t_bits[0]);
    tm1637.display(2, h_bits[1]);
    tm1637.display(3, h_bits[0]);
    float h = dht.readHumidity();
    float t = dht.readTemperature();
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.println(" *C");
}

#include "DHT.h"
#define BUZZER_PIN      36
#define TEMP_HUMI_PIN   24
DHT dht(TEMP_HUMI_PIN, DHT22);
int length = 2;
char notes[] = "gg";
int beats[] = { 1, 3 };
int tempo = 300;
void setup()
{
    dht.begin();
    pinMode(BUZZER_PIN, OUTPUT);
}
void loop()
{
  int temperature = dht.readTemperature();
  if(temperature>30)
  {
    for(int i = 0; i < length; i++) {
        if(notes[i] == ' ') {
            delay(beats[i] * tempo);
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        delay(tempo / 2);
    }
  }
}
void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(BUZZER_PIN, HIGH);
    delayMicroseconds(tone);
    digitalWrite(BUZZER_PIN, LOW);
    delayMicroseconds(tone);
  }
}
void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
  for (int i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}
```

*(Although I used Energia to cooperate with the LaunchPad, the codes are almost the same in either Energia or Arduino) The first code is to get readings from the sensor and divide the exact number into separate digits and then show them one by one on the 4-digit-display. Readings could also be observed from the serial monitor on Energia. The second is to make the buzzer work when temperature is above 30 degrees Celsius. The sound is designed to be: Notes: g, g; Beats: 1, 3; Tempo: 300ms.



IV. Results and Conclusion

1. Present Results and Analysis

Temperature and humidity are correctly shown on serial monitor and the 4-digit display. I tried to activate the buzzer by warming the sensor with hands to increase the temperature. However, hands easily increased humidity but not temperature. After I changed the threshold to be 20 degrees Celsius in the code, buzzer made its alarm.

The op-amp circuit, however, did not function well. There might be something wrong with the load cell, since the readings I got just fluctuated randomly and did not change according to the pressure on the sensor. This might be caused by large measurement noise of the load cell and I tried some method to deal with that but failed at last.

2. Lessons Learned

When considering adding a buzzer, my first idea was building a BJT circuit to control the voltage across a speak according to different analog on Arduino. However, I noticed that there was a buzzer in the Grove kit and it could be directly used on the LaunchPad. The rest would just be writing codes for its functioning. Using Arduino is much more convenient than hooking up some hardware circuit. I learned some Arduino language while doing the project, including some Arithmetic Operators and Compound Operators. For example, % (modulo) is used to calculate the remainder when one integer is divided by another. This is used in my code for 4-digit displaying, where the number of the value of temperature or humidity need to be divided into digits.

V. Future Work

The first step will definitely be to figure out how load cell functions with operational amplifier; perhaps I will try weight sensors of other kinds. After these sensors work in the expected way, the next step is to record several measurements and determine a relationship between its output voltage and the actual weight of objects.

In addition, the recording of temperature and humidity readings should be improved. The beekeepers cannot not always stare at 4-digit display or laptop screen; thus I need to figure out how to automatically calculate an average value each hour and save the data, which will be helpful for the final conclusion.

The buzzer might not be a good idea, since beekeepers are not near the hives for most of the time. I would better design some remote-communication system, like sending some signals to cellphones through Bluetooth or WiFi.

References:

Grove Starter Kit Manual

http://www.ti.com/lit/an/sloa034/sloa034.pdf

https://www.arduino.cc/en/Tutorial/Melody