Monitoring Honey Bee Hives Using Acoustic and Thermal Sensors

By Kristina Chu, Mihir Iyer, Kristi Lin, Paula Stocco, Xiang Wen (Thermal)

Mengxiong Liu, An Zhao (Acoustic)

ECE 110

James Scholar Honors Section

December 9, 2014

Table of Contents

Introduction

Problem description

For beekeepers, it is very difficult to track hive health during winter because when the colony is hibernating it is inadvisable to open the hive. During the winter, when the temperature drops to about 50 °F or below, honey bees retreat into their hive in order to conserve heat and protect the queen bee[1]. They form a cluster in the middle of the hive surrounding the queen, constantly fluttering their wings so the core of the cluster stays within 81-93 °F. Due to the vertical slat construction of the hives, the cluster is only able to move vertically within the hive, constricting the amount of honey they can consume. In order to reach the honey in the corners of the hive, the bees must break cluster, a process in which bees lose a significant amount of heat[2]. In a best case scenario, by the time the bees need to break cluster the weather will be warm enough for the bees to survive. However, this is not always the case. Our goal for this project is to design a sensor system that will be able to monitor the temperature and acoustic range of a hibernating honey bee hive. The system should be able to track the core temperature of the hive cluster as the winter progresses. A reach goal is to have the sensor see multiple heat spots within the hive to construct a rough estimate of the location of the cluster and see whether or not it has broken cluster. Additionally, the system should monitor the acoustic range of the hive. As the bees flap their wings during hibernation, the frequency they give off may differ in relation to environmental changes. By keeping a record of the frequencies, we will be able to see if and how they relate.

Design Concept

IR:

Our project monitors the movement of bees within the hive using an IR camera that tracks up to four heat sources or IR sources. The sensor takes advantage of the IR light the bees naturally emit/ the heat from the beating bee wings; seeing this heat source, the sensor will output a value to the Arduino, which will be accepted through a code. The data is converted into a visual representation by a second code, which illustrates position with four different colored

[1] http://en.wikipedia.org/wiki/Honey_bee#Winter_survival
[2] http://peacebeefarm.blogspot.com/2009/12/bees-make-cleansing-flights.html

dots on a black segmented background. The IR camera is powered by five volts (it has cables for ground and power). The camera requires an external IR light, that could be wired to the Arduino or an separate power source. Further testing and prototyping would determine the best light/sensor configuration. We suggest a square of lights with a hole in the center through which the fit the IR camera is pointed.

The third cable transmits gathered information. Currently this cable would physically connect the hive to the lab, preferably the Arduino will eventually wire to a wifi shield that would transmit the data remotely to the researchers[3]. When the bees break cluster, the distance between positioning points should increase. The scientists may monitor the distance between data points-- most likely at a greater frequency when the weather reaches a certain temperature or during the season the phenomena is usually observed. In the future, a code would be implemented to detect a preprogrammed distance and then alert the researchers to either physically view the high or monitor the information more closely.

Acoustic:

The sound of the animals has been a good source to monitor and judge the health and living conditions for animals for a long time. Especially for bees, the beeping sound can stand for many meanings. Different frequency can represent different ages and kinds of bees. The regular sound of "warming" vibration of bees and the signal of irregular beeping is what we want to concentrate on. In this way, we would be able to discover before the time that the bees tend to break the cluster.

Our project record the beeping sound of wings within the hive by setting eight microphones at eight corners of the hive. The microphone, which is the audio sensor, transfers the signal of sound waves into voltages, which is proportional to the intensity of sound. This voltage corresponds to the amplitude of sound waves. All the data will be dealt by sending all the data from the hive using radio or wifi to the computer.

---

[3]The effect of wifi on bees have been investigated, but is indecisive and contested.
http://ecowatch.com/2014/07/23/citizen-scientists-track-bee-health-colony-collapse-disorder/
The effect of IR on bees has also been documented, and bees appear to be unable to see IR light
http://www.bbka.org.uk/members/forum.php?t=2833

As the cluster has different distance from the corners of the hive, eight sensors will get similar sinusoid-like wave but shifted on the same time scale. By finding how much the sound wave has shifted on the time scale, the time difference between the signal of eight sensors is achieved. It is known that the speed of sound is constant in one medium, so the distance is proportional to the time sound wave traveled. The location of the cluster in the hive can be calculated due to these distances into a three-coordinate axis system, which is a similar idea to GPS.

Code will be used to calculate the frequency as well. As different frequency can represent different condition, we will statistic the regular condition for day period (which is the working time for bees) and the night period (when worker bees all stay in hive to keep temperature). In comparison of data, we will be able to figure out when the clusters are going to break out. (maybe like a high frequency sound or noisy wave).

Analysis of Components

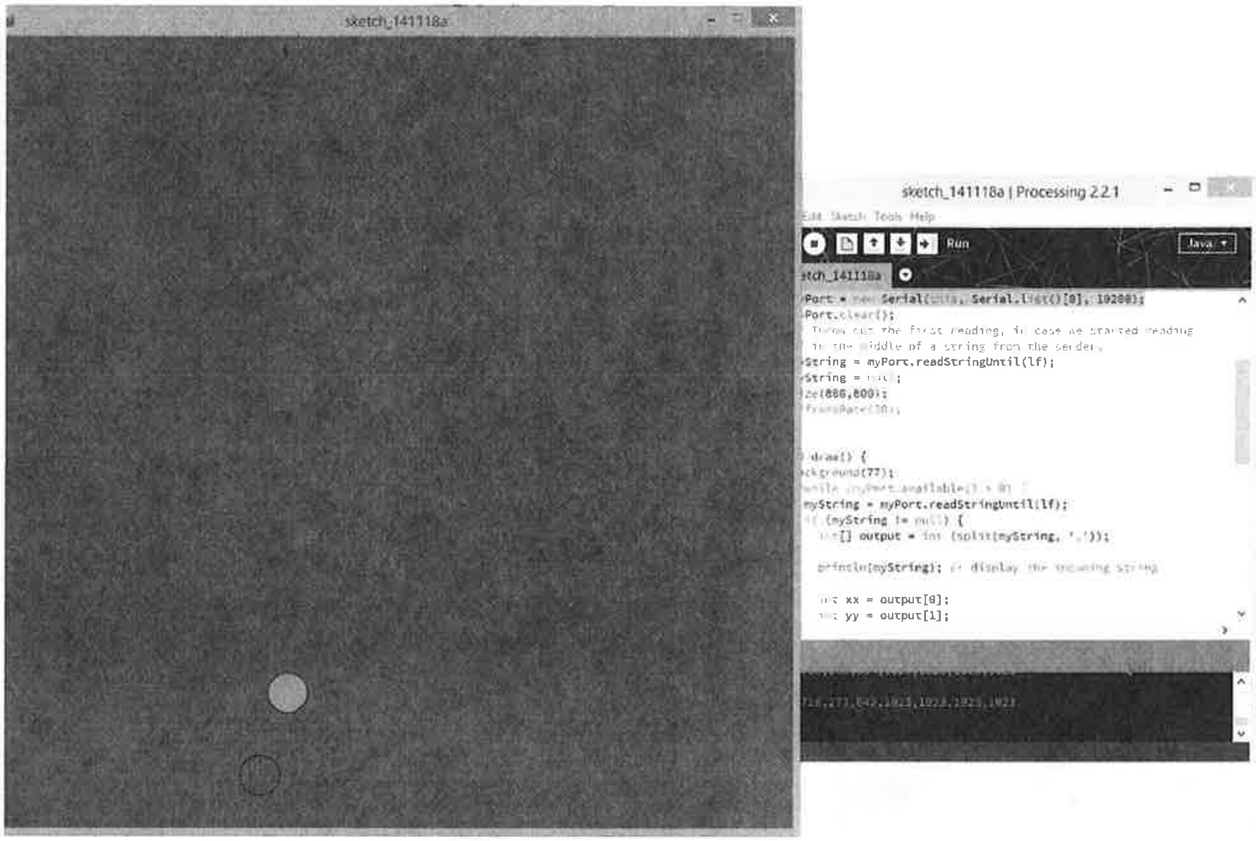
Characterization of Sensors

IR:

The goal that we hoped to achieve was a method by which researchers, beekeepers or any person interested in monitoring bees would be able to track the position of the bee clusters during cold seasons using thermal tracking. We were able to find an infrared camera that was capable of detecting multiple heat sources and feed that information to the Arduino microcontroller. Through coding with a program known as Processing, a program similar to the Arduino IDE that allows serial communication to the Arduino, we were able to create an x-y coordinate graph containing the detected heat source which showed up as a point on the graph. As the position of the heat source shifted, or if the camera was moved, the point displayed in the graph created by Processing would also shift correspondingly.

The camera is surprisingly powerful, being able to detect heat sources up to three meters away which is more than sufficient to monitor within a beehive. In addition, the field of view, 33 degrees horizontally and 23 degrees vertically, of the camera is large enough to look down the slot openings of the hive potentially making it the perfect sensor for our goal. Our initial experimental setup with the camera was a simple test of the functionality of the camera. The first step was to connect the camera into the four pins necessary to turn on the camera and communicate with the Arduino. The camera is powered by 5V and grounded. The transmission pins, SCL and SDA, allow the camera to communicate with the Arduino in serial. This process of communication is known the I2C (Inter-Integrated Circuit) protocol. We wired multiple infrared emitters to a breadboard and pointed the camera at the emitters. Sure enough, the camera was able to pick up the multiple infrared signals. In our code for the Arduino IDE, we were able to view the coordinates of the points on the graph. When no infrared signal is detected, the coordinates are set to the value 1023, the maximum value of the analogRead function. Once a signal is detected, the reading value fed by the camera to the Arduino decreases corresponding to the position of the detected infrared source.

Attached is a photo of the graph displayed by Processing. Like the code for the Arduino, Processing is also able to display the coordinates of the points. As shown in the picture, the camera was detecting two infrared sources which are displayed by the green and red balls respectively..



Our full codes for both the Arduino IDE and Processing can be found in the Appendix.

Acoustic:

The amplifying circuit of microphone is connected to the oscillator to test our sensor. As the bees will be really close to the sensor and the sound will be really loud, the data we achieved is using human voice 10-20 cm away from the sensor.

The sound wave appears to be very clear. The sinusoidal sound wave can reach the largest amplitude at about 349-352 mv, This is the clipping value of the amplifying circuit. As arduino deals with 0-5V and quantization for 7 bits, we only need to use 7 bit to quantize the sample as 352mv is less than ⅛ of 5 V.

However, sending out data to arduino appears to be not good. We found out that arduino is not fluent when dealing with audio, which is not a true A to D. We need to look for another device to transform our data from analog to digital.

Design Considerations

IR:

For the IR sensor, the main issue we encountered was that it only takes the temperature of what it first encounters. This means that if we put it between two of the hive slats, the sensor would only detect the temperature of the slats and not give a full thermal view of the cluster of bees. To account for this, we made the decision to mount the IR sensor on a track that will run back and forth across the multiple slot openings. We were provided with a wooden frame to be inserted in the top of the hive. That is where we will place the track. The sensor will detect the temperatures between each of the slot openings, and when put together will provide a comprehensive of the temperature gradient within the hive.

Additionally, another major concern was that bees secrete a type of sticky substance called propolis that covers any object within 1 cm they come into contact with[4]. This 1 cm area around a bee is called bee space. If we placed the sensor in bee space, the bees would gunk it up with propolis, interfering with its sensing capabilities. In order to combat this, we decided to house the IR sensor track behind a mesh screen. The mesh screen will be built into the wooden insert, and will protect the sensor from propolis.

Acoustic:

The first issue we dealt with is we need to amplify the signal achieved by sensor in order to input the value into an A to D device. To account for this, we built an amplifying circuit. We first used a variable resistance from 10K-1M to adjust the value and look for a resistance that suits best for us. We found out even with 1 Mega ohms, the gain is still too little to satisfy our requirement. In this status, we built a second amplifier to amplify the signal again.

[4] http://www.honeybeesuite.com/bee-box-terminology-what-to-call-all-the-sizes/

The second concern is that how should we avoid the noise created by bees crush or touching directly onto the microphone. A good way would be adding something between the bees and the microphone, which is also a regular way in our daily used microphones. Cotton material would be wrapped around the sensor to decrease volume and noise, and a hard net cover would be added to decrease the vibration of bees of directly touching or crashing.

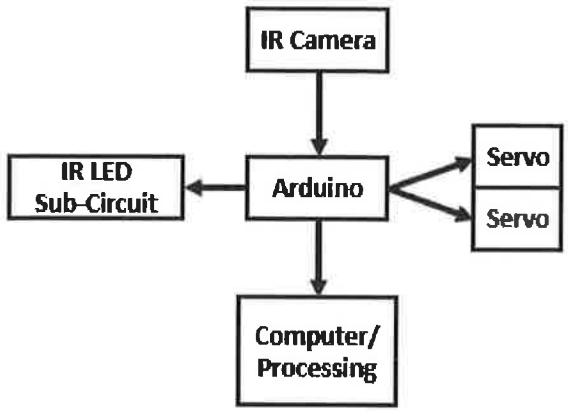
Then we thought about how should we plot a graph of the 3 coordinate axis system of the location of the cluster. With all the datas would be collected, matlab would be a good way.

Another major concern is that we cannot really use wire to "transport" the signal back to our computer as it is too limited for us to deal with large amounts of data. We need to use someway that is wireless to remote the operation of sensor. Building a remoter and adding signal transmitter onto the sensor is preferred by us.

Design Description

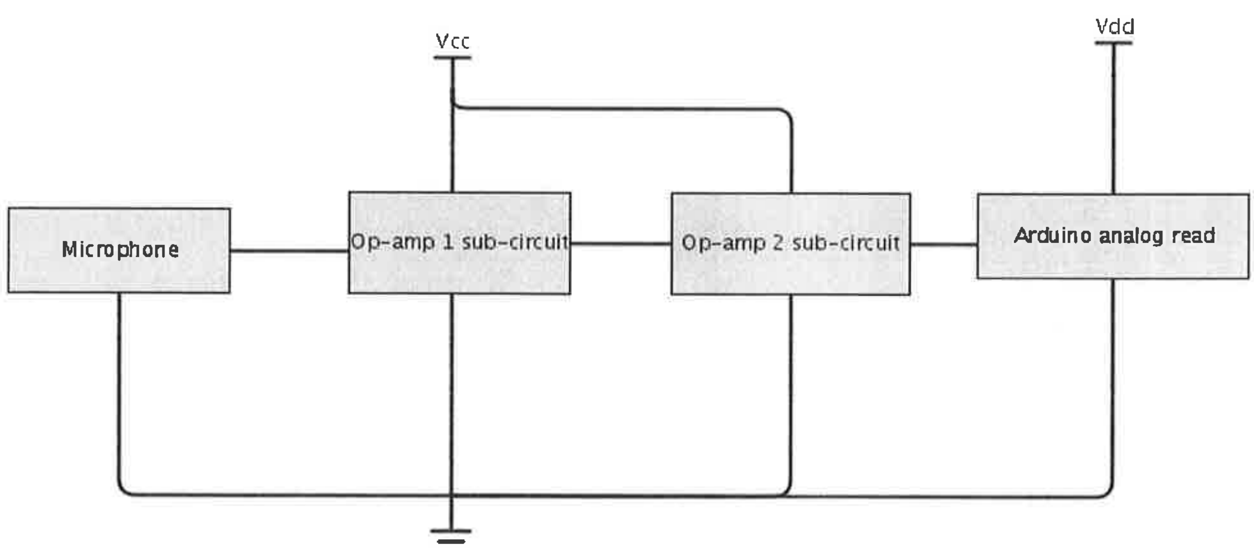
Block Diagram

IR:



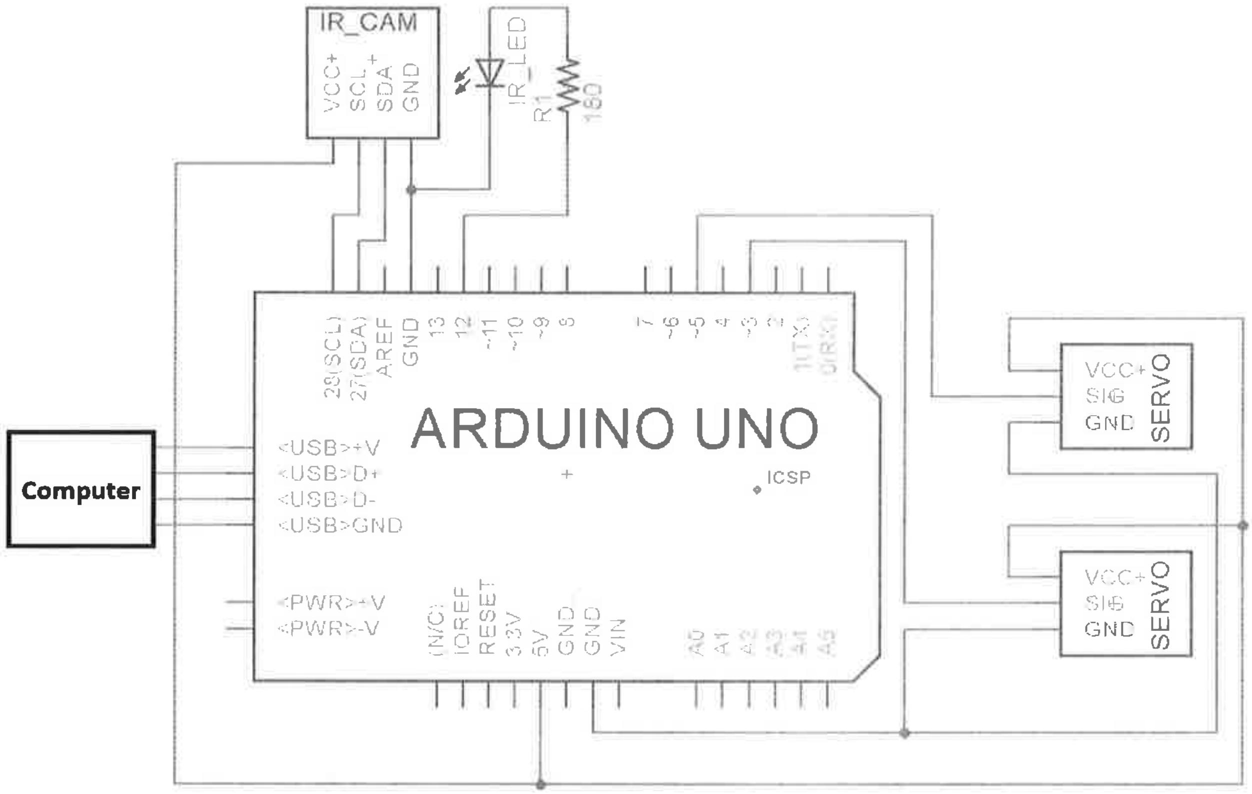
From this block diagram, it is clear that the IR Camera is the only input in this system. The Servos and the LED Sub-Circuit do not depend on the information from the camera, but they are still controlled by the Arduino. The information from the camera is sent from the Arduino to the computer which is running Processing in order to generate a plot of the positions of IR sources in the camera's field of vision.

Acoustic:

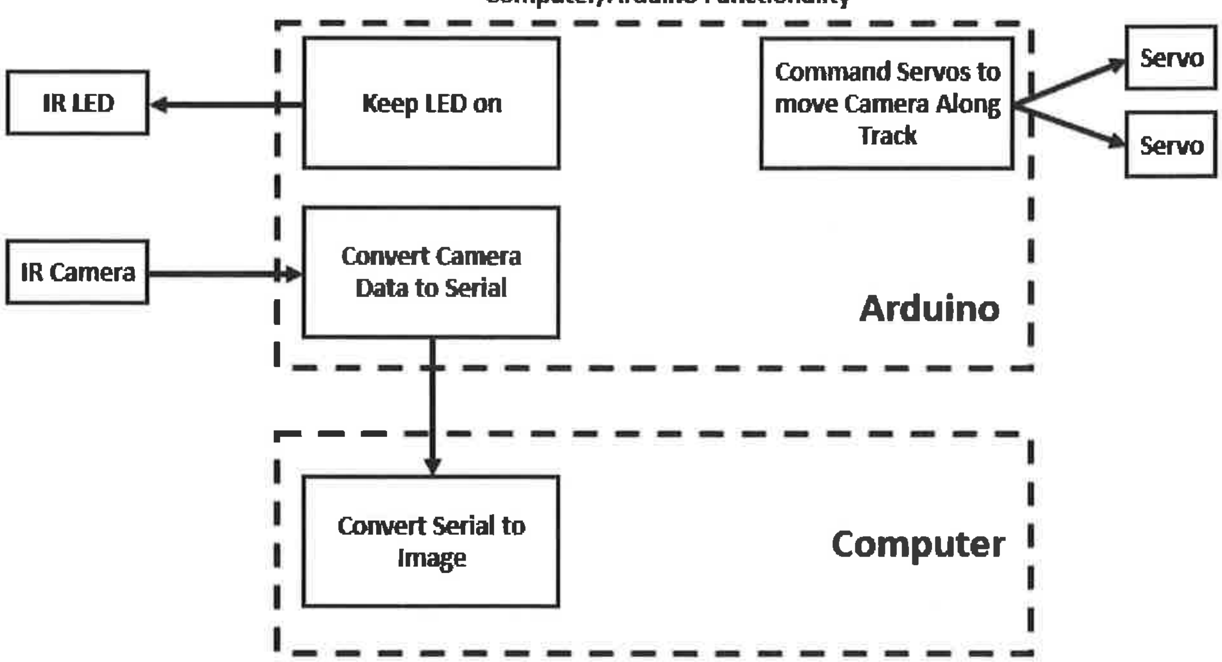


Circuit Schematics

IR:



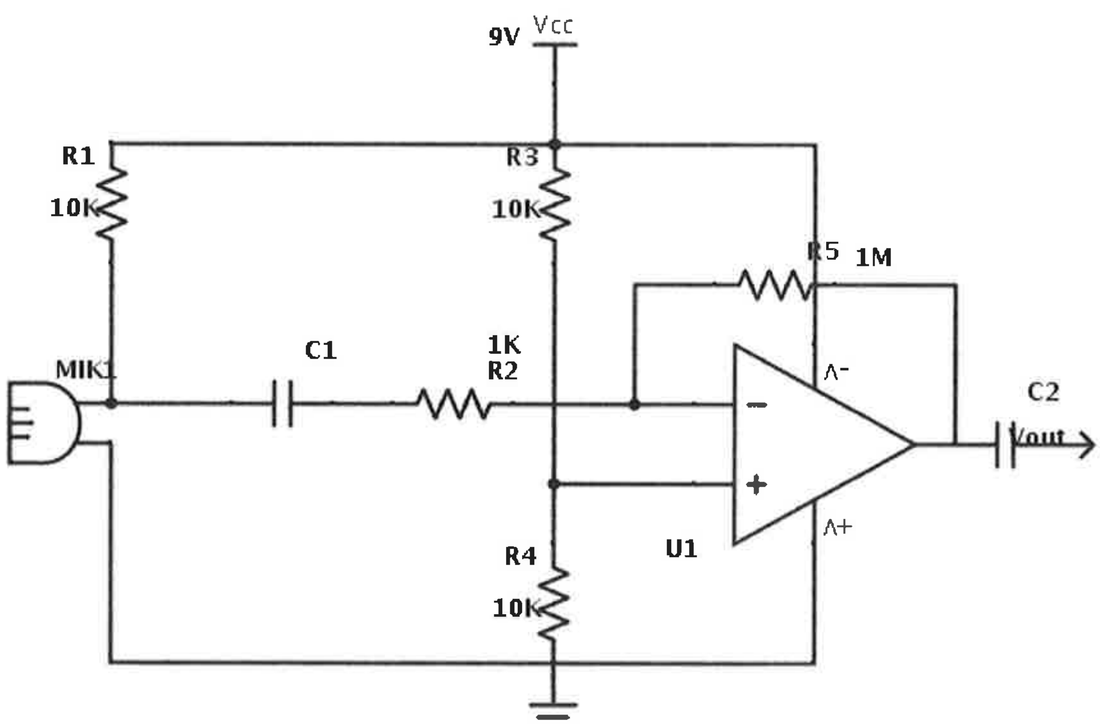
Computer/Arduino Functionality



On the right side of the schematic, we can see the servos. They receive power and ground from the Arduino and are controlled using PWM. Because of this, they have to be connected to special PWM capable digital I/O pins on the Arduino which are denoted by a ~. In this case we use pins 3 and 5. We then program the Arduino to generate PWM signals which drive the servos to different positions. The next part of the circuit is the IR LED and resistor. The resistor limits the current so we do not damage the LED. The IR camera communicates with the Arduino using I2C so we must use the SDA and SCL pins. The information from the camera is then sent over serial to the computer which is running a Processing program that convert this data into a plot of the locations of IR sources.

Acoustic:

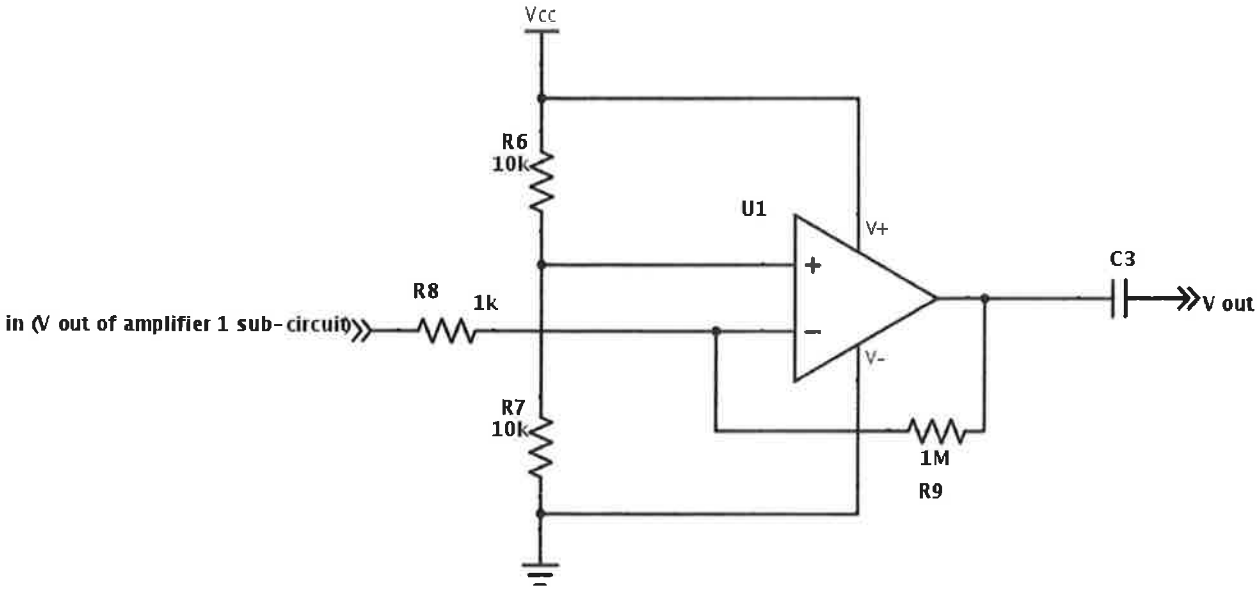
Microphone op-amp 1 subcircuit



Vout = – V_IN * R5 / R2= -V-IN * 1000

as capacitance C2 is blocking DC circuit, Vcc does not increase Vout.

Op-amp 2 sub-circuit



This second amplifying circuit increase the peak value above the original clipping value so that a larger amplitude can be achieved and analyzed.

Physical/mechanical construction

IR:

The IR sensor, arduino, IR light, and power source would sit on a shelf of a special research row built for the beehive, and view down into the slots. A separating thin screen between the shelf and the honey-combs would prevent the bees from coating components with propolis.

The IR sensor detects horizontally at 33 degrees, and vertically at 22 degrees, and at distances from 0 to 3 meters.[5] The optimal IR face angle would best see as much of the slot volume as possible, and would probably be best decided by prototyping on the research board, and angling the sensor until the most clear data is obtained.

To cover the entire beehive width, without multiple IR sensors and data screens, the IR sensor will scan the box back and forth. An automation system would move the IR sensor

[5] This is from the provided data sheet. http://www.dfrobot.com/wiki/index.php/Positioning_ir_camera
When a sensor is bought, it should be BE CHARACTERIZED because rarely is a physical product exact to its product specifications (it may be found very off)

horizontally across the board, ideally avoiding shaking the box excessively, maintaining the IR sensor at its optimal angle, and capable of reversing directions.

Here we propose multiple solutions.

IR sensor and IR light can be mounted onto a cart[6] to run along a track at the edge of the shelf. Servos on either wall would pull the cart back and forth by a string. Arduinos can be used to provide the pulse width modulation voltage for the servo, however the appropriate pulse train[7] will need to be tested for its exact value to be found. Again, specifics depend on the exact models and characterization for servo, cart, track, string.

Another option would be to still mount the IR camera and IR light to a cart, but run the cart through a lead screw[8], which would be turned from an external motor. However, the motor required might be larger, the movement less accurate, and the integration of a mating nut into the cart difficult.

Designs will evolve with continued testing.

Acoustic:

The microphone with remote-receiver and signal transmitter will sit on the same shelf as thermal sensors on the top part of the beehives, at 4 corners. At the bottom 4 corners, the sensors will stick onto the bottom shelf, hide inside the edges of the hive at 4 corners. The microphone will be covered with cotton material and a net cover to prevent extra noise from bees stepping onto the microphones directly.

All 8 microphones will be facing the inside of the hive, at angle about 35 degrees from horizontal, towards the center of the hive. The best choice to minimize the volume of our device is to build everything onto a piece of PCB.

---

[6] Such as a toy train set! Prototyping doesn't require expensive equipment!
[7] Pulse train = multiple voltage pulses (square wave)
http://www.sciencebuddies.org/science-fair-projects/project_ideas/Robotics_ServoMotors.shtml?from=Blog
[8]A little background on how Lead Screws work
http://www.sdp-si.com/Acme-Lead-Screws.htm

Conclusions

Lesson Learned

IR:

Throughout the course of this project, our team has learned many things. Going into this project without much information on honeybees and their lifestyles made it difficult to even know where to begin. As mentioned in the introduction, our task was to find a way to monitor the health of a hive during the winter. More specifically, we decided we wanted to monitor the temperature inside of the hive and to ideally be able to track the movement of the cluster using strips of some kind of thermal sensor to position the heat source. By implementing a system which can take this into account, we believed that we would be able to obtain a better grasp of how much honey the hive consumes based on the movements of the cluster and that we would be able to learn more about the what actually happens inside the hive during the winter months.

The first lesson we learned involved picking the sensor. Since none of us have had experience with monitoring hives and we had determined that we wanted to monitor the temperature, we began by looking into thermal cameras, but this quickly proved to be a problem, seeing that many of these cameras were outrageously priced. Much of our project was spent looking into different type of thermal sensors and comparing their cost versus their accuracy and their different capabilities.

After looking even further, the TMP006 chip, which is used in the TI SensorTag, was suggested to us. From reading more about this it seemed perfect. Ideally it could accurately detect the temperature from the distance across the hive, it was small and it seemed like it would not be too difficult to program. After borrowing the SensorTag to see how it worked, we were slightly disappointed on how it worked. The sensing distance was not nearly what it claimed to be, so this would not work to give us an accurate reading across the whole hive. From this,we were able to learn to always check what the companies claim about their sensors and the accuracy. Although the datasheet could say one thing, the sensor may not always work as planned, so it is always good to test it out.

Next, we found Thermal Ribbons, that would be able to give us a view of the heat in the hive and the relative temperatures in various spots if we were to line them up on the sides of the wall. After researching further, however, we discovered that if we wanted to obtain an accurate reading we would need higher numbers of these strips than initially expected, making them not very cost effective. This taught us another valuable lesson. In today's world, money really limits the possibilities, and often as price increases, so does accuracy and quality of data. This relationship between the price of the sensors and the accuracy is a key thing to consider when choosing a sensor. The key is finding the threshold where the sensor is accurate enough to obtain the needed data, but it is not so expensive as to empty one's wallet.

The sensor we ended up purchasing was an Infrared Heat Source Tracker, that seemed to do exactly what we wanted according to the website where we found it. Although it does not record temperature, it is able to track the movement of four different heat sources and theoretically should be able to track the cluster within the hive. As with our other sensors, we ran into some trouble. The sensor used Arduino in conjunction with Processing, and no one in our group knows Processing, which made it difficult to understand what exactly the code was supposed to be doing. The website provided us with code, however, when initially it did not seem to work, we did not know how to debug it. This is a very important thing to consider when purchasing a sensor. Always knowing how it works and what it is supposed to do can help a lot when attempting to characterize the sensor. It would also have been beneficial if we has investigated the sensor more and had more information about it before purchasing it. We ran into another problem with this sensor after looking more into similar sensors and discovering that we needed an IR emitter. Once we used an IR emitter, our sensor was able to track the movement, however, without pointing it at the IR emitter, we were unable to get it to work properly. While bees do emit IR light, we are still unsure whether they will emit enough for our sensor to pick up, especially since we were unable to get the sensor to work on humans as the heat source it was supposed to track.

In summary, from all of our issues with sensors, we learned that the three most important factors are to consider when choosing a sensor are cost, quality of data received and available knowledge about how the sensor works. Choosing a sensor that fits all of the desired criteria can be difficult,especially without much knowledge about what is out there that would allow us to accomplish our goals.

If we could do this again, we believe that we would have to spend more time figuring out this sensor, or actually trying other sensors and determining which works the best instead of spending so much time researching. The most valuable part of this was to actually work with the sensors, and it gave us the clearest idea of which sensor would suit our needs, as we did with the TI SensorTag. Being able to experiment more with this sensor and a real hive would also be beneficial, so we could see if bees do in fact emit IR light, or if we were to put an IR emitter coupled with our sensor to measure the distance at which the cluster is, it would be good to test and see if bees are affected by high amounts of IR light. Now that there is some groundwork laid for this project, we can only hope that future honors sections can learn from what we did well and from what we wish we had done differently and create even more successful systems to monitor the health of honeybee hives.

Acoustic:

The main goal of our team, as mentioned above, is to use the microphones to locate the cluster. Although finally we only partly finish the circuit, we still learnt a lot from this experience.

The first thing we learnt is how to use an op-amp. We learnt about the inner structure of op-amp, how the output voltage depend on the two input voltage and what an op-amp circuit should look like. However, before we pick a specific type of op-amp, we did not gather enough data to deeply analyze how much "gain" we need and what the clipping voltage should be. Therefore, when we mount one op-amp and analyze the signal, we found out that the maximum voltage is not high enough. It is simply around 500mV while we expect our max plug voltage to be nearly 5V, which is the maximum Arduino analog read voltage. So we had to mount another op-amp to amplify our "gain" again. Therefore, in our future study, we will collect sufficient data and do theoretical analysis before we pick the components we use.

In addition, we learnt how to analyze complicated circuit. The circuit diagram seems to be a mess at first glimpse. We had a hard time understanding the circuit because we did not understand the functionality of each component. However, after we figured out the functionality of each component, the circuit becomes much easier to analyze. For example, we learnt that the gain is the the ration of R5 and R2 so that we can adjust R5 and R2 to control our gain. Also, we learnt that C2 is used to resist DC, which means removing the effect of noise, and that R3 and R4 are used to maintain the one input voltage to be half of Vcc. With these understanding, we can adjust our circuit the way we want instead of "copying" the circuit from the graph on our breadboard.

Self-Assessment

IR:

Looking back on what we have done, we did not accomplish as much as we hoped we would have. Aside from not beginning to meet until about halfway through the semester, we spent probably longer than we should have looking at and researching different sensors. In reality, the most practical course would have been to pick something that may not have been what we were looking for exactly and characterized it quickly, instead of doing countless hours of theoretical work through research. Once we were able to try out a few sensors, we learned what we liked and did not like, and we were made aware of aspects to watch out for.

When we finally ordered a sensor we did not have enough time to figure out how to use it properly and actually get it working as we thought. We encountered some problems with our sensor not working how we thought, and since the semester was nearing its end, we did not have time to solidify our solutions to these problems, or to find a new sensor that may have worked better. Although we did get our sensor working in the end, we still are unsure if it will work as desired if it is implemented into a hive.

Another thing that we were unable to do was to actually create a design for how we will mount the sensor. We were able to discuss our design and we believe that it would work, but we were unable to physically make it. In the future, once the sensor has been chosen, it would be better to have two people work on the sensor itself and have two people work on how it would be mounted.

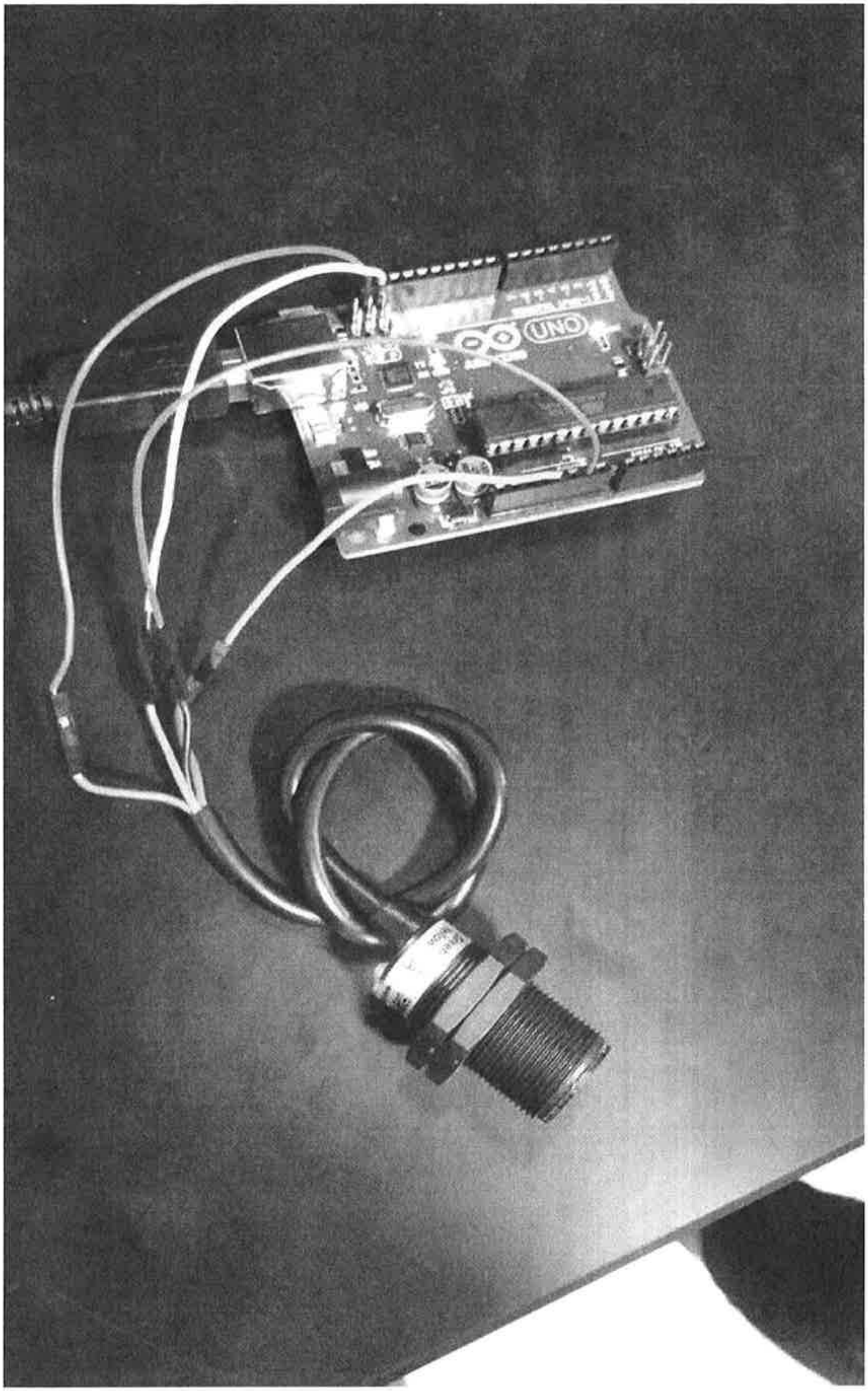
Despite the fact that this project did not go as well as we had initially hoped, we were still able to learn a lot about the engineering design algorithm and about problem solving in general.

This project gave all of us the valuable experience of working independently on a different sort of project and will allow us to avoid the same mistakes in other project similar to these. Because of this,we will know how to solve problems and overcome obstacles that may appear in a wide scope project like this. We also learned how to work with each other in a group, which will be very beneficial as we further our engineering education. As the concept for this project is just emerging, we realize that this is not the end, its just the beginning. Best of luck to those continuing on with this project.

Acoustic:

Our current progress is about half way to our designated goal. The main inefficiency comes from our disorganization. Before we start to build out circuit, we spent about two weeks searching data of the bee hives and four weeks looking for the sensors we will be using and corresponding circuits. We found most circuits to be complicated without even trying to analyze or build them. However, when we finally decide to build a circuit that seemed to be complicated, we do not have much difficulty to make it work. We learnt that we should play with the seemingly difficult stuff before we give up.

Second, we learnt to finish our task step by step. Actually, before we even start our circuit, we consider what to do with the signals processed, including how to use matlab to transfer time domain signals to frequency domain signals. We once considered to use this method to calculate the phase difference (the time delay) between several signals. We hoped to use this time difference to calculate the distance difference of the cluster to each microphone. However, this turns out to be out of our ability. So instead, we decide to use a simpler model, measuring the intensity difference of each signal to locate the cluster. The signal analysis consideration do not help a lot when we actually start building our circuit. We should have only focused on the basic concept of data analysis and leave the main calculation consideration after we finish our circuit because. If we do it step by step and concentrate on the circuit first, we can save a lot of time.


UNO

Appendix

IR:

Processing code:

```
import processing.serial.*;

int lf = 10;    // Linefeed in ASCII
String myString = null;
Serial myPort;  // The serial port

void setup() {
 // List all the available serial ports
 println(Serial.list());
 // Open the port you are using at the rate you want:
 myPort = new Serial(this, Serial.list()[0], 19200);
 myPort.clear();
 // Throw out the first reading, in case we started reading
 // in the middle of a string from the sender.
 myString = myPort.readStringUntil(lf);
 myString = null;
 size(800,800);
 //frameRate(30);
}

void draw() {
 background(77);
 //while (myPort.available() > 0) {
  myString = myPort.readStringUntil(lf);
  if (myString != null) {
   int[] output = int (split(myString, ','));

   println(myString); // display the incoming string

   int xx = output[0];
   int yy = output[1];

   int ww = output[2];
   int zz = output[3];

   int xxx = output[4];
   int yyy = output[5];

   int www = output[6];
   int zzz = output[7];

   ellipseMode(RADIUS);  // Set ellipseMode to RADIUS
   fill(255, 0, 0);  // Set fill to white
   ellipse(xx, yy, 20, 20);
   ellipseMode(RADIUS);  // Set ellipseMode to RADIUS
   fill(0, 255, 0);  // Set fill to white
   ellipse(ww, zz, 20, 20);
```

```
      ellipseMode(RADIUS);  // Set ellipseMode to RADIUS
      fill(0, 0, 255);  // Set fill to white
      ellipse(xxx, yyy, 20, 20);
      ellipseMode(RADIUS);  // Set ellipseMode to RADIUS
      fill(255);  // Set fill to white
      ellipse(www, zzz, 20, 20);

    }
}
```

Arduino IDE code:

```
#include <Wire.h>

int IRsensorAddress = 0xB0;
//int IRsensorAddress = 0x58;
int slaveAddress;
int ledPin = 13;
boolean ledState = false;
byte data_buf[16];
int i;

int Ix[4];
int Iy[4];
int s;

void Write_2bytes(byte d1, byte d2)
{
  Wire.beginTransmission(slaveAddress);
  Wire.write(d1); Wire.write(d2);
  Wire.endTransmission();
}

void setup()
{
  slaveAddress = IRsensorAddress >> 1;   // This results in 0x21 as the address to pass to TWI
  Serial.begin(19200);
  pinMode(ledPin, OUTPUT);      // Set the LED pin as output
  Wire.begin();
  // IR sensor initialize
  Write_2bytes(0x30,0x01); delay(10);
  Write_2bytes(0x30,0x08); delay(10);
  Write_2bytes(0x06,0x90); delay(10);
  Write_2bytes(0x08,0xC0); delay(10);
  Write_2bytes(0x1A,0x40); delay(10);
  Write_2bytes(0x33,0x33); delay(10);
  delay(100);
}
void loop()
{
  ledState = !ledState;
  if (ledState) { digitalWrite(ledPin,HIGH); } else { digitalWrite(ledPin,LOW); }

  //IR sensor read
  Wire.beginTransmission(slaveAddress);
```

```
Wire.write(0x36);
Wire.endTransmission();

Wire.requestFrom(slaveAddress, 16);        // Request the 2 byte heading (MSB comes first)
for (i=0;i<16;i++) { data_buf[i]=0; }
i=0;
while(Wire.available() && i < 16) {
  data_buf[i] = Wire.read();
  i++;
}

Ix[0] = data_buf[1];
Iy[0] = data_buf[2];
s  = data_buf[3];
Ix[0] += (s & 0x30) <<4;
Iy[0] += (s & 0xC0) <<2;

Ix[1] = data_buf[4];
Iy[1] = data_buf[5];
s  = data_buf[6];
Ix[1] += (s & 0x30) <<4;
Iy[1] += (s & 0xC0) <<2;

Ix[2] = data_buf[7];
Iy[2] = data_buf[8];
s  = data_buf[9];
Ix[2] += (s & 0x30) <<4;
Iy[2] += (s & 0xC0) <<2;

Ix[3] = data_buf[10];
Iy[3] = data_buf[11];
s  = data_buf[12];
Ix[3] += (s & 0x30) <<4;
Iy[3] += (s & 0xC0) <<2;

for(i=0; i<4; i++)
{
 if (Ix[i] < 1000)
  Serial.print("");
 if (Ix[i] < 100)
  Serial.print("");
 if (Ix[i] < 10)
  Serial.print("");
 Serial.print( int(Ix[i]) );
 Serial.print(",");
 if (Iy[i] < 1000)
  Serial.print("");
 if (Iy[i] < 100)
  Serial.print("");
 if (Iy[i] < 10)
  Serial.print("");
 Serial.print( int(Iy[i]) );
 if (i<3)
  Serial.print(",");
```

```
  }
  Serial.println("");
  delay(100);
}
```