Mosquito Trap

May 5, 2010

Team:
Jeff Theisen – Team Leader
John McGuire – Communicator
Courtney Krueger – BWIG
Ryan Nessman – BSAC

Client:
Dr. David Van Sickle
Department of Population Health Sciences

Advisor:
Dr. Chris Brace
Department of Biomedical Engineering

Abstract

It is important to be able to trap and analyze mosquitoes in order to determine what steps should be taken to prevent the transmission of the diseases they carry. Current mosquito traps require an incredible amount of time and tedious work for entomologists. A mosquito trap which implements remote transmittance of data could be extremely useful, as it would not only decrease the amount of work for entomologists significantly, but it could also provide real-time data of the environmental conditions when mosquitoes are entering the trap. Three additions to the current trap were integrated: sensing (the ability to count the number of mosquitoes in the trap), communication (wirelessly transmitting the data) and differentiating and speciating the mosquitoes. A GSM module was be used as the communication device, audio frequency detectors were chosen to carry out differentiation and speciation aspects, and a laser tripwire was implemented to sense entering insects.

Problem Statement

This project is designed to apply a variety of electronic technologies to current light trap models of mosquito traps to improve their utility and ease of use and their potential to generate valuable, timely data for public health. In particular, we plan to integrate a variety of sensors to collect data on the local environment and on the trapped mosquitoes, and provide a means to remotely monitor and operate the device.

Background

General Purpose

Mosquitoes are capable of carrying deadly diseases, such as malaria, and transmitting those diseases to humans. This transmission has been a major health issue throughout the world; an estimated 300 million people are infected by malaria each year and over 3 million of these individuals die [1]. There are over 3,000 existing species of mosquitoes in the world today, but only a fraction of these are capable of spreading disease [2]. In an effort to control the mosquito's spread of disease, entomologists make use of mosquito traps, which allows them to monitor the populations of various species of mosquitoes [3]. Using the data received from the traps, entomologists are able to make accurate and informed decisions regarding mosquito control.

Current Traps

In an effort to receive the best data, entomologists make use of two popular kinds of mosquito traps, the gravid trap and the light trap. The light trap (shown in figure 1) attracts mosquitoes using carbon dioxide and light, while the gravid trap (shown in figure 2) attracts mosquitoes using swamp water. The different methods of attraction found within the traps are used to attract different species of mosquitoes. In the light trap, the carbon dioxide is emitted from a canister which is filled with dry ice, and the light is emitted from an incandescent light bulb [3]. In the gravid trap, the swamp water rests in a pan [4]. Aside from their method of attracting mosquitoes, the gravid trap and the light trap are essentially identical. Once mosquitoes are attracted to a certain point, there is a plastic tube which has a fan inside of it. The fan is turned on when the trap is in use, and mosquitoes are sucked through it and pushed through the cylinder and into a bag, where they stay. The bag's only entrance is through the cylinder, and the fan is powerful enough to keep the mosquitoes from flying out [4]. In order to keep larger bugs out of the traps, there is a mesh filter which is large enough to let mosquitoes pass through but stops the larger bugs. These traps are powered by a standard 6V battery, which typically holds enough power to run for one night. To use the traps, entomologists place them in a designated area late in the evening, where they run for the duration of the night. They then come and collect the traps in the morning, and proceed to hand count how many mosquitoes they collected. The entomologists then speciate each individual mosquito under a microscope, based on their scaling patterns on the abdomen and striping patterns on the legs. This allows the entomologists to receive the data that they need to make informed decisions.

Figure 1: Light Trap [5]

Figure 2: Gravid Trap [6]

Design Motivation

The overall motivation behind our project is to enhance the existing mosquito traps to make the process of trapping much more automated. Because entomologists have to individually count all of the mosquitoes that enter the trap and then individually speciate them under a microscope based on their scaling patterns, the process of mosquito trapping takes a vast amount of time. In one trap, there can be up to 65,000 mosquitoes in one night [7]. Usually, entomologists use multiple traps at once, so using the traps consumes an enormous amount of time, as scientists have to potentially speciate thousands upon thousands of mosquitoes. Despite the fact that entomologists usually take a subset of the data, which usually consists of about 2,500 mosquitoes, that is still an incredibly large number of mosquitoes to have to count and speciate [4]. Given this, the main motivation behind this project is to decrease the amount of man-hours needed to accurately monitor mosquito populations within the area. Other problems with the current design are that it does not have temperature or humidity sensors, which would be useful for obtaining information about the local environment of the trap. It also has no way of communicating remotely and no way of establishing real time data. If these given problems were solved, monitoring the populations of mosquitoes would be made much easier and more effective, while the amount of man hours devoted to the monitoring would be significantly decreased.

Client's Requirements and Design Constraints

The client provided several specifications that the final design must meet. First among these, was that the device must be durable and rugged, and withstand weather and exposure. The trap will be operating in an outdoor environment; thus, it must be able to endure various weather conditions such as rain, wind, dirt, or dust. It should withstand temperature ranges from 50°F to 115°F and showers of rain. Current light traps are designed with an umbrella-like, protective disk on top, which will offer some shielding from the weather.

Another specification is that the device must reliably capture and count an accurate and valid sample of mosquitoes of interest. This is so that the trap can provide quality data to entomologists who make decisions based on this data. The main requirement is that the trap must provide data which correlates to the amount of mosquitoes in the trap; for instance, if the trap were to detect only 20% of the mosquitoes, but do so reliably night after night, this would give the entomologists a good idea of how many mosquitoes are present at any given time. This data should be communicated remotely, in order to provide real-time information. Additionally, the trap should limit the number of other insects counted and captured, as the mosquitoes are the insects of interest.

Several operational specifications were also set, to ensure that the trap will be able to operate in a satisfactory manner. One of these is that the electronic additions must not diminish the length of trap operation. Current traps operate for the entire length of a night, so the device must be able to last for this amount of time. Since there are batteries for use with mosquito traps that have lives up to 20Amp-hours, our trap must consume less than that amount of power in a single night. The designed device must also be relatively simple to operate, so as not to make it confusing or difficult for those who want to use it. Furthermore, the device could be designed as an add-on unit to current traps.

Finally, the client also provided several production characteristics. The device is relatively unrestricted in size and weight, except that it must be small and light enough to facilitate easy transportation to and from a mosquito collection site. This will be accomplished by limiting the size to less than 2ftx2ftx4ft and the weight to less than 40lbs. The client desires just one prototype, which will operate alongside current traps this summer for comparison. Additionally, the budget was set at $1000 for development of this device; however, the cost of reproducing the final product should be significantly less. Current traps range from $90 to $300 plus the cost of batteries and chargers, which could easily add $100 [8]. The cost of the final device could be slightly greater than these values, as it will offer a more automated, efficient process of mosquito collection.

The formal product design specifications, which contains all of this information in a more itemized format, can be found in the appendix.

Design Alternatives

Through meetings with our client, discussions with entomologists, and team brainstorms, a list of about 15 potential improvements was generated. Some of the proposed improvements include: regulating and automating the release of carbon dioxide, scavenging power through solar panels, killing mosquitoes as they enter the traps, incorporating temperature and humidity sensors, remotely differentiating mosquitoes from other bugs, remote control and retrieval of information, and many more. The team narrowed this list down to manageable set of priorities to be addressed in this semester; these goals reflect a desire to communicate useful data about the set of mosquitoes entering the trap, and do not focus on attraction methods or dealing with the bugs after they enter. Priorities:

1) Accurately detect and count insects entering the trap.
2) Remotely communicate this data.

3) Improve the quality of data by potentially differentiating mosquitoes from other bugs or speciating the mosquitoes.

After a brief summary of constant features of the design, the following sections of design alternatives will address these priorities.

<u>Constant Features of the Design</u>

There are a number of aspects to this design which were not considered with alternatives because they we relatively easy to decide upon. The first being the microcontroller implemented to interface and control the various electronic devices. For this, the client provided us with the Arduino Diecimila, as seen in figure 3. This board has 14 digital input/output pins, 6 analog input pins, 16kb of flash memory, a clock speed of 16 MHz, and can be program easily using the Arduino programming language. It optimally runs on 7-12 volts DC, although it can power by as low as 6 volts DC [9].

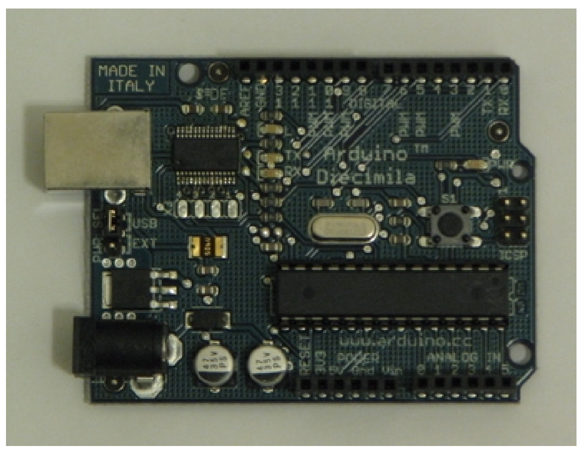

Figure 3: Arduino Diecimila [9]

The next constant feature is the waterproof case in which the electronics will be housed, thus keeping them safe from the weather. Although a final case has not been selected, several good possibilities have been identified, one of which is shown in the figure 4 below, which has inner dimensions of 5.79 x 3.11 x 2.4 in. [10]. The final decision will be made once the optimal dimensions needed to house all the electronics are know.

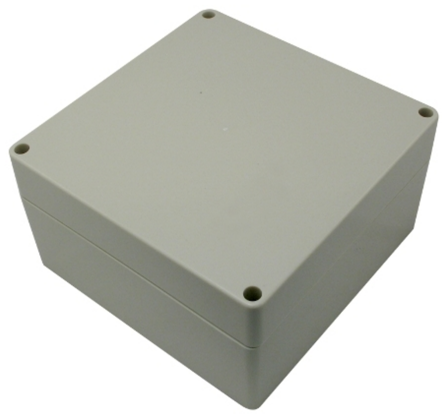

Figure 4: Waterproof plastic electronic enclosure [10]

The final constant feature of this design is the power supply. For this, the team will incorporate the battery currently used by the client to power the mosquito traps. These are 6V batteries with 10Amp-hours of battery life. This will make it easier on the client, because they already have the charger that works with this battery. As mentioned early, the Arduino runs optimally on 7-12 volts; operating at 6 volts may cause the board to be unstable [9]. If, through testing, this is determined to be a problem, several approaches could be used to find a solution. The Arduino could alternatively be supplied by two of these 6V batteries in series, effectively powering it with 12V. Additionally, similar 12V batteries can be purchased, though these would require a new charger as well [8]. As described in the client requirements section, the trap must operate through the entire night. Most of the electronics we plan to incorporate have relatively low power consumption; the fan draws approximately 300mA, the Arduino draws about 32mA, the GSM draws about 60mA, and other components will consume even less. However, if additional battery life is needed, more batteries can be purchased from John W. Hock with battery lives of up to 20Amp-hours [8].

<u>Design Aspect 1: Sensing</u>

The first design aspect is sensing, that is, the ability to detect and potentially count mosquitoes once they have entered the trap. There have been methods used to count insects using sensing devices, including a laser trip wire and an infrared motion sensor [11]. These two implementations will be tested and the method that is most effective will be used in the final design.

Laser Trip Wire

The first method examined was the laser trip wire. This device uses an infrared beam generated by a photodiode (figure 5), which is focused on a photo resistor. The photo resistor senses the infrared light from the laser, and a resistance is established. Once an object (in this case, a mosquito) crosses the beam, the beam is broken. This causes the resistance within the photo resistor to change. The change in resistance is transmitted to the Arduino device, signaling that something has broken the beam [12]. The Arduino would then be able to record and transmit the number of times the beam was broken, thus giving a count of mosquitoes.

The laser trip wire is advantageous because it detects a break in the laser beam, which could be caused by something as small as a mosquito. The main concern with sensing is the ability to detect and count mosquitoes, and this implementation has a large probability of being successful.

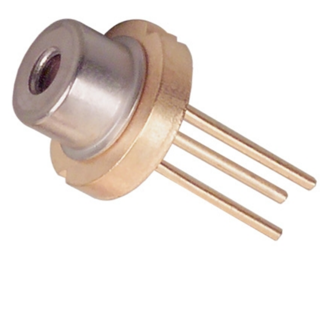

Figure 5: Image of a laser diode [13]

The main problem with this design is that, because it uses a single laser beam, it may not be possible to cover the entire area of the trap. It could be difficult to sense mosquitoes, as they could pass through the trap and not break the beam. It is possible, however, to implement a series of mirrors within the trap to reflect the beam, and increase the amount of space that it covers. Alternatively, the mosquitoes could be forced through a smaller passage, through which the laser passes, in order to increase the possibility of breaking the beam. One source of error, is that it is also possible for the beam to be broken by other insects, and potential debris that passes through the trap, which would result in and inaccurate mosquito count. This error is currently minimized by physically limiting the insects that enter using mesh, and plans to incorporate a differentiation method will further reduce this error.

Motion Sensor

The motion sensor mechanism is able to detect the motion of infrared sources. The device uses a dome-shaped lens to detect infrared radiation, with an optical filter place behind it, as diagrammed in figure 6. The motion of an infrared source can be detected by these components through a transfer of heat from the source to the filter. The change in heat is transmitted through the device, and amplified to the output [14].

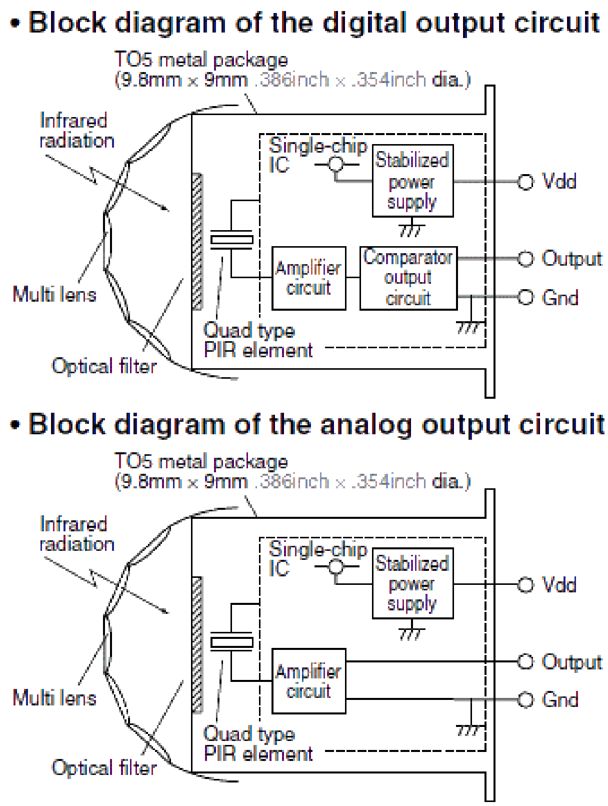

Figure 6: Diagram of the motion sensor mechanics [14]

The main advantage to this method is that it can cover a broad area, and can easily detect infrared sources within the trap. Mosquitoes emit infrared radiation, and could be detected using such a method.

The key disadvantage of this device is that it is designed to detect infrared radiation from a human source, which at normal body temperature radiates a wavelength of approximately 12 micrometers. The infrared radiation emitted by mosquitoes is around 6-8 micrometers, so the device would need to be modified in order to detect that wavelength. It is also uncertain as to how the device will react to multiple infrared sources.

Design Matrix for Sensing

The design matrix for sensing is not complete yet, as testing the parts has not yet been conducted. Three factors will be taken into account as to which method will be used:

effectiveness, ease of use with Arduino, and cost. Each design will be given a rating based on their performance in each of the three categories.

The heaviest weighted criterion of this design is effectiveness, which is its ability to sense mosquitoes. Due to the need for a count of mosquitoes within the trap, it is essential to be able to sense the mosquitoes and produce a count. The laser trip wire and infrared motion sensor will be tested and their performance will be rated based on the ability to sense mosquitoes.

It is also essential for the sensing device to be compatible with the Arduino microcontroller. In order for our device to transmit data, it must first pass through the Arduino. In turn, it is essential for all electronic elements of the trap to be compatible with the Arduino.

Cost always weighs in when considering the possible designs, as it is ideal to make the most inexpensive functioning prototype. The components for the laser trip wire cost $17.00, making it relatively inexpensive, as the components for the infrared motion sensor cost $44.24. Because the infrared motion sensor components cost $44.24, which is not expensive considering the budget of $1000, it received 8 out of 10 possible points. The laser trip wire components cost only $17.00, so it was awarded 10 out of 10 possible points.

Table 1: Design Matrix for Sensing

Considerations	Weight	Motion Sensor	Laser Tripwire
Effectiveness	70	TBD	TBD
Ease of Use with Arduino	20	TBD	TBD
Cost	10	8	10
Total	100	TBD	TBD

Design aspect 2: Communication

The second aspect examined for improving the mosquito trap was communication. In order to achieve the goal of transmitting real-time data from the trap, it is necessary to implement a device that can take the signals from the readings within the device and transmit them to a recording mechanism. Methods for transmitting data from an Arduino exist, including a GSM transmitter, and wireless local area network (wifi). These two methods were evaluated based on their effectiveness and efficiency within a mosquito trap.

GSM

The first possible method to be used in the mosquito trap was a GSM module. This device is commonly used in devices such as cellular phones for transmittance of data, live audio, and video recordings. GSM uses technology that transmits the signal to a satellite, where it is reflected to a destination specified on the transmitter. In this case, it would allow for the data from the trap to be transmitted to a receiver that is connected to a computer, where it can be analyzed [15].

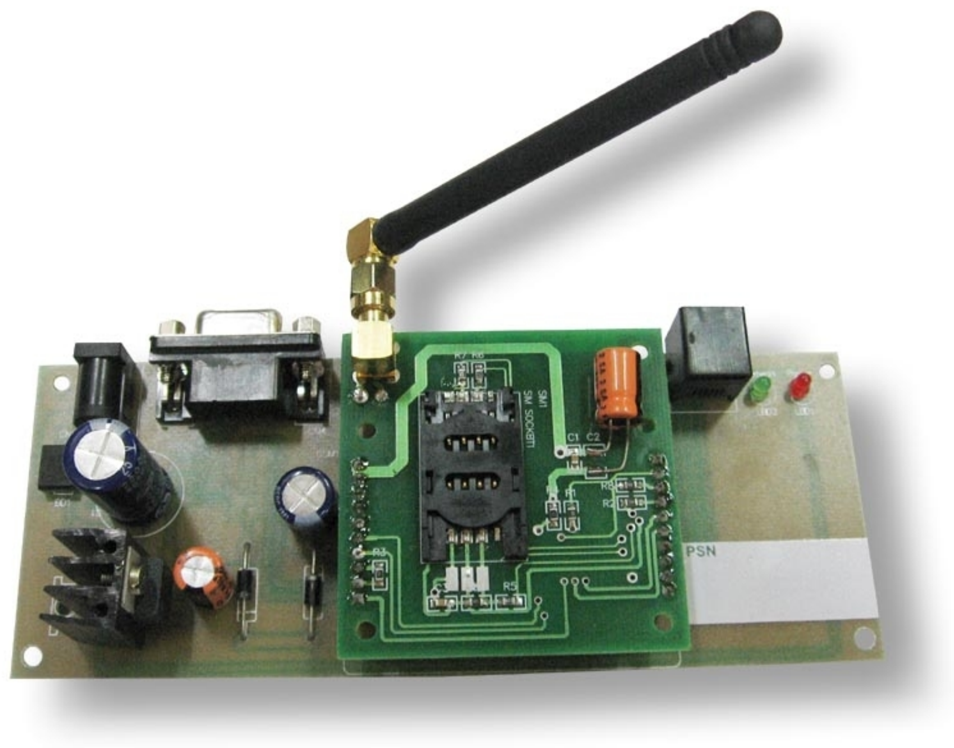

Figure 7: Image of a GSM transmitter [15]

GSM transmission is advantageous because it is a proven method of long-distance transmission. Devices such as cellular phones that have GSM modules are able to transmit signals from many remote locations, as long as there is GSM coverage in the area. It is common for GSM to have a large coverage area, as many cellular phone companies want their devices to be able to operate in the largest area possible. In the case of the mosquito trap, there are a number of wireless companies that have coverage in all the areas where the traps are placed. GSM is also easily compatible with the Arduino microcontroller, and the mechanism comes fully capable of transmitting data, so no extra construction would be necessary.

The disadvantage of using GSM to transmit data is that it could be expensive. Wireless providers that facilitate the transmission of data require payment for the amount of data being transmitted. In the case of the mosquito trap, it would be necessary to transmit data and perhaps audio recordings from the device, and it could prove to be very expensive. It is undetermined exactly how much it would cost to transmit the data from the trap to a recording device, however a typical wireless provider charges about $40/month for unlimited audio transmission.

Wifi

The second option for transmitting data is wifi. Wifi is commonly used to provide wireless internet to a specified area. It allows for access to online transmission and reception of data, including audio and video. Wifi uses radio signals that can be analyzed and converted at both an internet source and the transmission device to allow for data transmission [16].

Figure 8: Image of a wifi transmitter [16]

The key advantage of using wifi is its cost-effectiveness. One flat rate would allow for unlimited transmission of data from the source. For an internet connection, a standard rate is approximately $20/month. A wifi receiver located on the device would come fully assembled, and would be easily implemented in collaboration with the Arduino microcontroller, making it easy to assemble.

The main disadvantage of wifi communication is its range. Commercial wifi sources are designed to provide service to areas such as single buildings, and are not yet capable of having a range that spans a large area. Because the mosquito traps are sometimes set in rural areas, it would be very difficult to obtain wifi signals. It could be possible to establish wifi signals through an internet provider at areas where the traps would be used, however, this would be very costly.

Design Matrix for Communication

The design matrix for communication was based off of four aspects: cost, ease of construction, ease of use, and range. The wifi and GSM methods were evaluated based on their performance in those four categories.

The cost of transmitting data from the trap is important because the device should be as efficient as possible, and if the components of the trap are too expensive, it would be difficult to implement the design. The cost of GSM transmission of data would be in the area of $40/month for data transmission. This is fairly expensive but could be accomplished with the current budget for the trap, so it received 17 out of 25 possible points. Because wifi communication would cost approximately $20/month for data transmission, it received 22 out of 25 possible points, making it the method of choice in this category.

Both GSM and wifi communication would come fully assembled, making them almost effortless to assemble. Because they both would require such little effort, they each received 15 out of 15 possible points in the ease of construction category.

Ease of use is essential when determining which method of communication to use. The entomologists must be able to easily operate the devices and troubleshoot problems within the mechanism. Each communication method is fairly easy to use, as they are currently used in many transmission devices. Because both devices require a similar amount of knowledge to operate, they both received 25 out of 30 possible points in the ease of use category.

The final aspect examined in the design matrix was range. The range of the device was the key factor in deciding which device to implement, as it is essential for the device to be able to transmit data from the remote locations where the traps will be placed. Because GSM has the capability to transmit data from a large area of locations, it received 29 out of 30 possible points. Wifi fell behind in this category because it has a very limited range and can only be implemented in locations that have wifi connections. Because of its limited range, wifi received only 10 points out of a possible 30.

After the points were totaled, the GSM module came out on top, with 86 out of 100 potential points. Although it lost points in the cost category, it was not as crucial as the range aspect, in which it received the most points by far. Wifi received 72 out of a possible 100 points, mainly because of its lack of range, which proved to be the deciding factor in making the decision. Because GSM scored the most points overall, it was chosen to be implemented in the final design.

Table 2: Design Matrix for Communication

Considerations	Weight	GSM	Wifi
Cost	25	17	22
Ease of Construction	15	15	15
Ease of Use	30	25	25
Range	30	29	10
Total	100	86	72

Design Aspect 3: Differentiation and Speciation

For the purpose of counting and speciating mosquitoes, it is important that the trap is able to remotely differentiate mosquitoes from other bug and/or speciate the mosquitoes that are entering the trap. There are some insects that can enter the trap that are not mosquitoes, as they are a similar size to mosquitoes and can therefore get through the mesh openings. Although many of the other insects that try to enter the trap are too large to get through the mesh, there are still a fair number of insects that are very similar to mosquitoes that do not transmit disease and therefore are unimportant to the researcher. These insects vary in their population numbers during different times of the year, so it is not feasible to extrapolate the number of mosquitoes from the number of total insects caught. Under a microscope, which is the current method used by researchers to differentiate and speciate mosquitoes, it is very easy to recognize another insect that is not a mosquito. It is important to speciate the mosquitoes because different species of mosquitoes are capable of carrying diseases, while others are just considered pests. The information gathered by researchers about the numbers of different species of mosquitoes helps mosquito control groups determine if they need to deploy mosquito control methods, such as still-water draining or spraying.

There are three methods for differentiating/speciating mosquitoes. One method involves taking pictures of each entering mosquito. If this picture were to be sent to the researchers and it was of high enough quality, then they would be able to quickly and remotely differentiate and speciate the mosquitoes by looking at their scales, relative size, and leg bands. However, they normally view the insects underneath a microscope, so the picture that was sent would have to be very high resolution. In order to do this, we would need to obtain a very small, high resolution camera that took a picture when the counting sensor was set off. It might have to autofocus, although we would attempt to keep the insects in a small focus area by narrowing their entrance path considerably. The light conditions inside the trap would require a flash to be used. All of these aspects would make the picture method a finicky design to have to work with, and its accuracy might be limited by the quality of the picture that it would be able to take. The cost of this method would also be high because small, high-resolution cameras are expensive and would be likely to make the cost of each trap high.

The audio frequency method is based off of the fact that different insects, species of mosquitoes, and even the different genders of the same species of mosquito, flap their wings at a different frequency from one another. If the sound of their flight is recorded with a microphone and this data analyzed, strong correlations between frequency and species can be determined and used to speciate each mosquito[17]. For the best data, a short recording should only be saved and sent as a file to the remote computer if the count sensor is tripped. The computer can then further analyze the data to determine the frequency that was heard. Advantages of this method include the ease of implementability, the relative simplicity of construction, and cost. A microphone with a high enough quality for this function should be easy to find; it would need a range of 100Hz-4kHz to detect mosquito frequencies. Some problems that will need to be worked out include filtering out the background noise from the wing beat frequency signal, but computer software or operational amplifiers should be able to do this fairly easily. This method has also been used in several other mosquito studies with 90% accuracy [17].

The light frequency method works off of the same idea as the audio frequency method. A laser light is shown onto a photoresistor. When an insect is in the beam of the laser, the resistance pattern on the photoresistor changes and includes information about the frequency of the wing beats. Looking into this method however, it is apparent that there are a large amount of parts required, such as photomultiplier tubes and lenses, that would require large amounts of power and proper positioning, respectively. [18] This would make the setup unable to be applied to different kinds of existing traps. The software required by the computer would also be fairly complex. For these reasons, the light frequency method was not determined to be a feasible idea that we could implement within one semester's time.

Design Matrix for Differentiation and Speciation

In the end, we chose to focus on the audio frequency method for differentiating and speciating the insects that enter the trap due to its implementability, ease of data transmission, and differentiation ability. In the design matrix, all methods received low scores for speciation due to the assumption that it would be much easier to differentiate mosquitoes from other insects than to speciate the mosquitoes. A raw count of mosquitoes in itself would be very valuable to the researchers, even if we could not end up speciating the mosquitoes from one another.

Table 3: Design Matrix for Differentiation and Speciation

Considerations	Weight	Pictures	Audio	Light
Implementability	40	35	35	30
Data Transmission	10	2	6	5
Differentiation	30	29	28	15
Speciation	20	6	7	5
Total	100	72	76	55

Final Design

In the final design, depicted in figure 9, the mosquito will be attracted to the trap from a container full of stink water. It will be sucked up by the force of the fan, through the mesh, and will be blown up the large PVC tube into the holding container. On its way up the tube, it will pass through a funnel, where it will be counted with the counting sensor and the frequency of its wing beats will be recorded by a microphone. The count and frequency data will be sent with Arduino through the GSM module to the computers at the entomology lab where it can be further analyzed. All of the electronic parts will have waterproof casing to protect them from the outdoor elements. They will be further protected by a rain shield that is placed on top of the trap. Design elements that have stayed constant are the use of the an existing gravid trap, a 6 volt battery provided by the client, and the attraction methods.

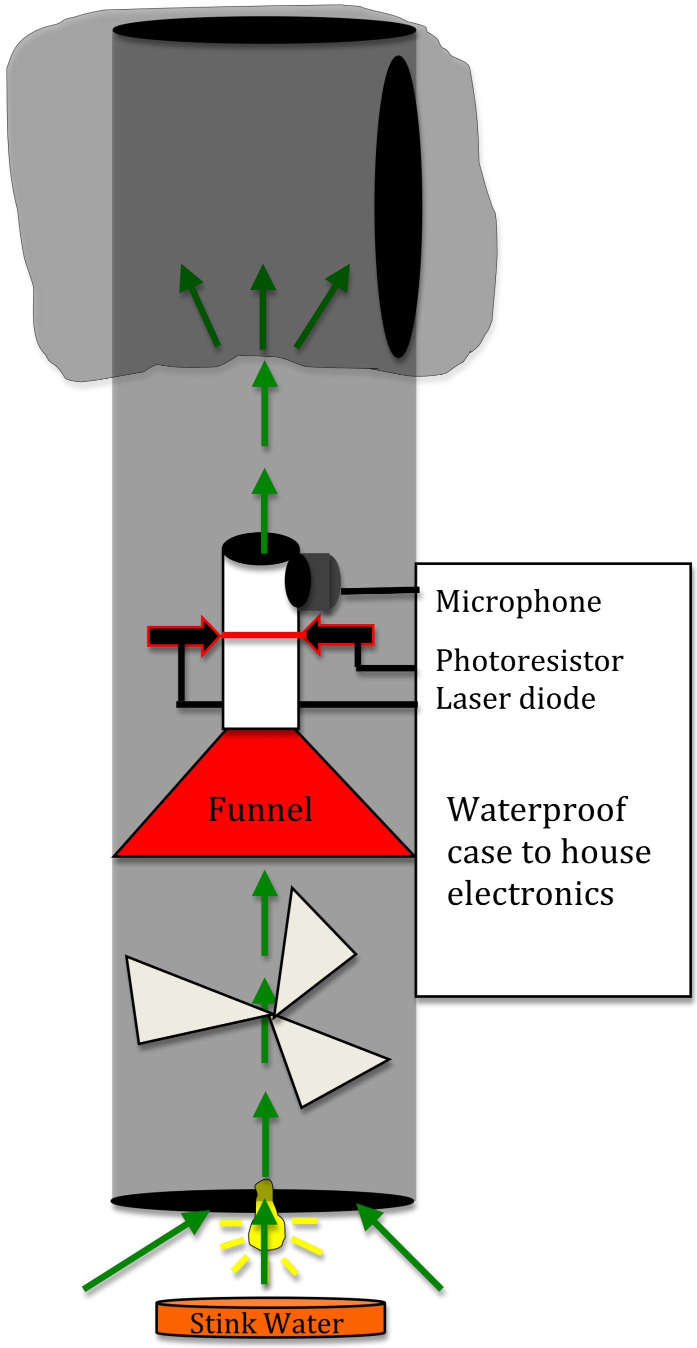

Figure 9: Final design for the mosquito trap

Circuit design

The circuitry of the final design consists of a variety of elements, all of which are controlled via the Arduino. These elements include a the fan, which is toggled via a MOSFET switch, the audio amplifier, which is connected to the microphone and is also toggled via a MOSFET switch, the GSM module, which is connected to a DC/DC converter to regulate the voltage, the photoresistor night sensor, the laser diode focused on the photoresistor, and of course the Arduino. A block diagram schematic depicting these connections is shown in figure 10.

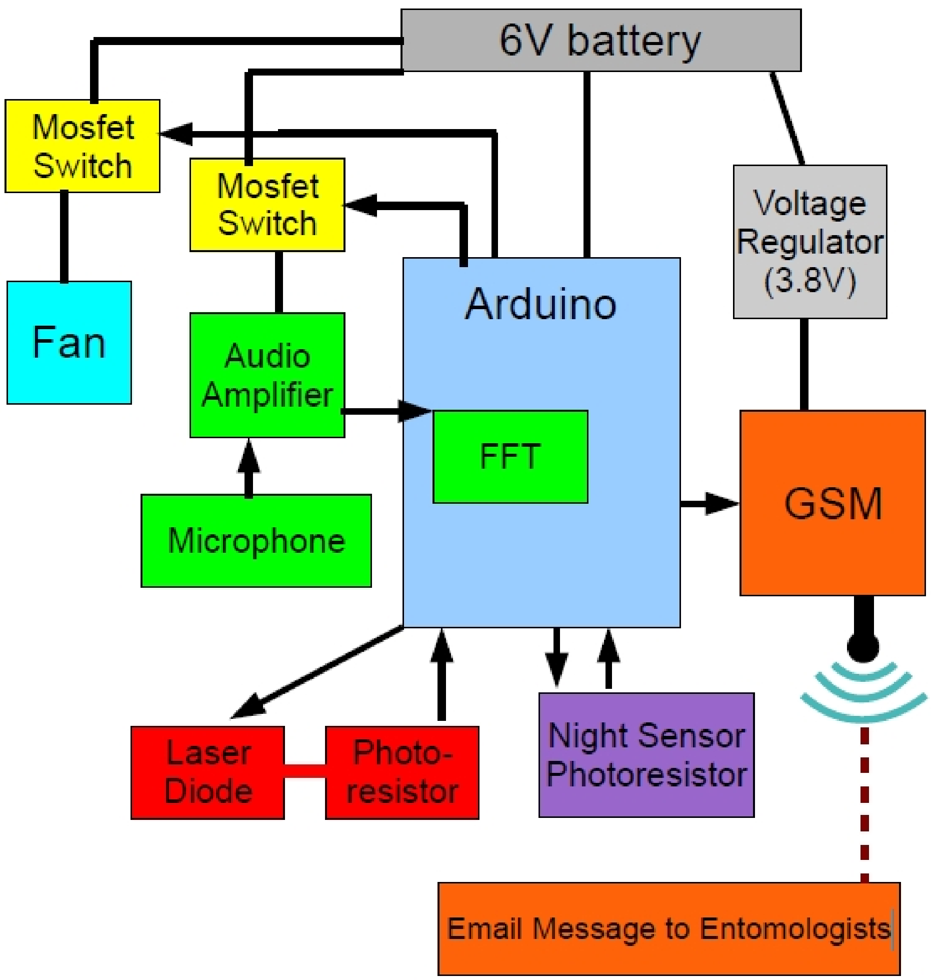

Figure 10: Block diagram of circuitry

Laser Tripwire

The laser tripwire has two main components, the laser diode and the photoresistor. These components are shown in the circuit in figure 11. The laser diode used in this design (Lumex

Opto/Components Inc: OED-LDP65001E) emits red light at 650nm and operates optimally at 25mA; the 120 ohm resistor in series with the laser limits the current to this optimal value. The photoresistor (PDV-P5002, made by Advanced Photonix Inc.), was found to have a resistance of about 400-600 ohms when illuminated by the laser diode; thus, a 680 ohm resistor was placed in a voltage divider with it, providing minimal power consumption and near optimal resolution of resistance changes in the photoresistor. The Arduino reads the voltage across the photoresistor. When a mosquito breaks the beam, a shadow is cast on the photoresistor, increasing its resistance, thereby increasing the voltage read by the Arduino.

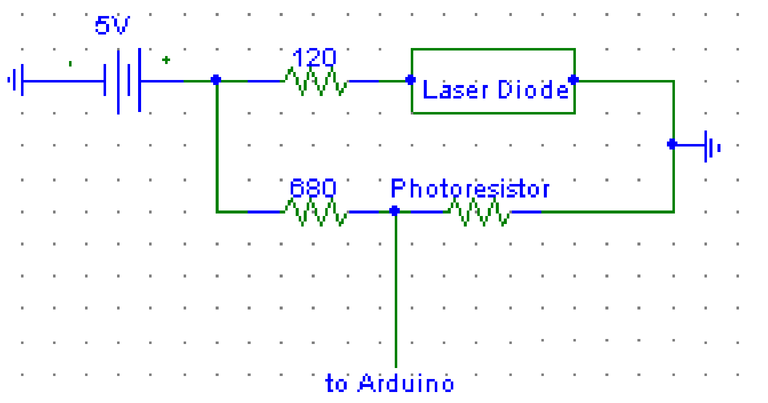

Figure 11: Circuit diagram of the laser tripwire

Microphone and Amplifier

The design utilized an electret condenser microphone (Panasonic: WM-55D103), which needs to be biased and amplified to obtain quality signals. The need for biasing turned out to be advantageous because the Arduino can read voltages between 0V and 5V, so the bias keeps signals in these ranges. The amplifier circuit uses the LM386 audio amplifier, the circuit for which was obtained from an the catalog drawings [19]. This circuit, with a few modifications to fit our needs, is shown in figure 12. A passive low pass filter was added to this amplifier circuit to minimize interference; the corner frequency for this was designed to be 10kHz, allowing the desired frequency range of 0-4kHz pass. Also a voltage divider was introduced to drop the 6V source to the needed 1.5V bias voltage for the microphone, although in the future this voltage divider should be replaced with a voltage regulator to reduce power consumption.

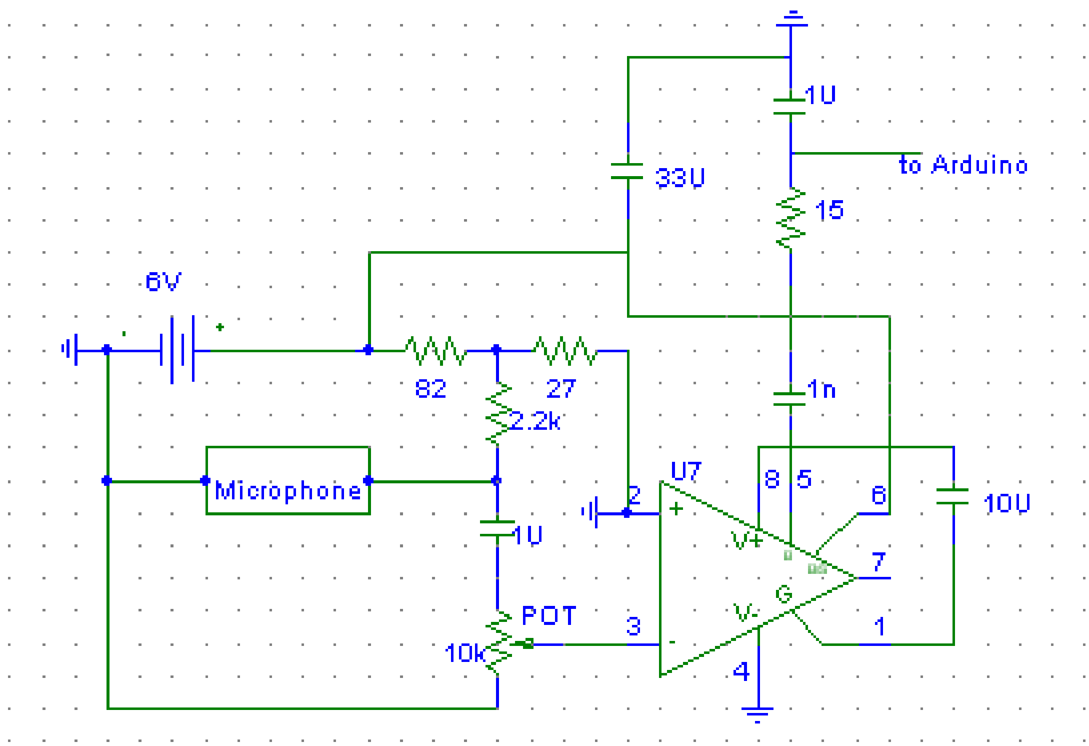

Figure 12: Audio amplifier circuit diagram

Since the current draw from the audio amplifier circuit was 65mA, a MOSFET switch was used to toggle this amplifier on and off, which allows for power conservation. The MOSFET circuit, using the IRF520NPBF MOSFET made by International Rectifier, is depicted in figure 13. In this circuit, rather than connecting the audio amplifier straight to ground, it was connected to the MOSFET, which was controlled from the Arduino, to allow current to flow through the load only when desired [20].

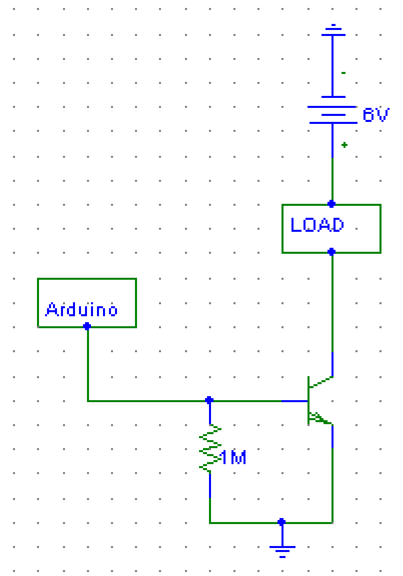

Figure 13: MOSFET switching circuit, used with the audio amplifier and the fan

Fan

The fan currently in use by the clients was the one used in the final design. This fan draws approximately 300mA when connected to the 6V battery. To minimize power consumption, the fan was incorporated into another MOSFET switch circuit, constructed in the same manner as described for the audio amplifier (shown in figure 13).

GSM

The circuit used to communicate with the GSM module was found on a project website for interfacing Arduino with the GM862 [21]. This circuit is shown in figure 14. The connection between pin 2 of the Arduino and on/off of the GM862 is used to tie the GSM pin to ground, allowing it to turn on and off. Pins 3 and 4 on the Arduino (which are TX and RX) go to TXD and RXD on the GM862, respectively; these connections allow for serial communication between the Arduino and GSM. The 100 kohm resistors are to pull up the voltage, since the pins are not internally pulled up, and the voltage divider on the TX line is to bring the voltage to the necessary 2.8V required for the CMOS logic on the GSM. The status LED provides valuable information about the GSM, such as whether or not it is connected to the network, based on the frequency of flashes [22].

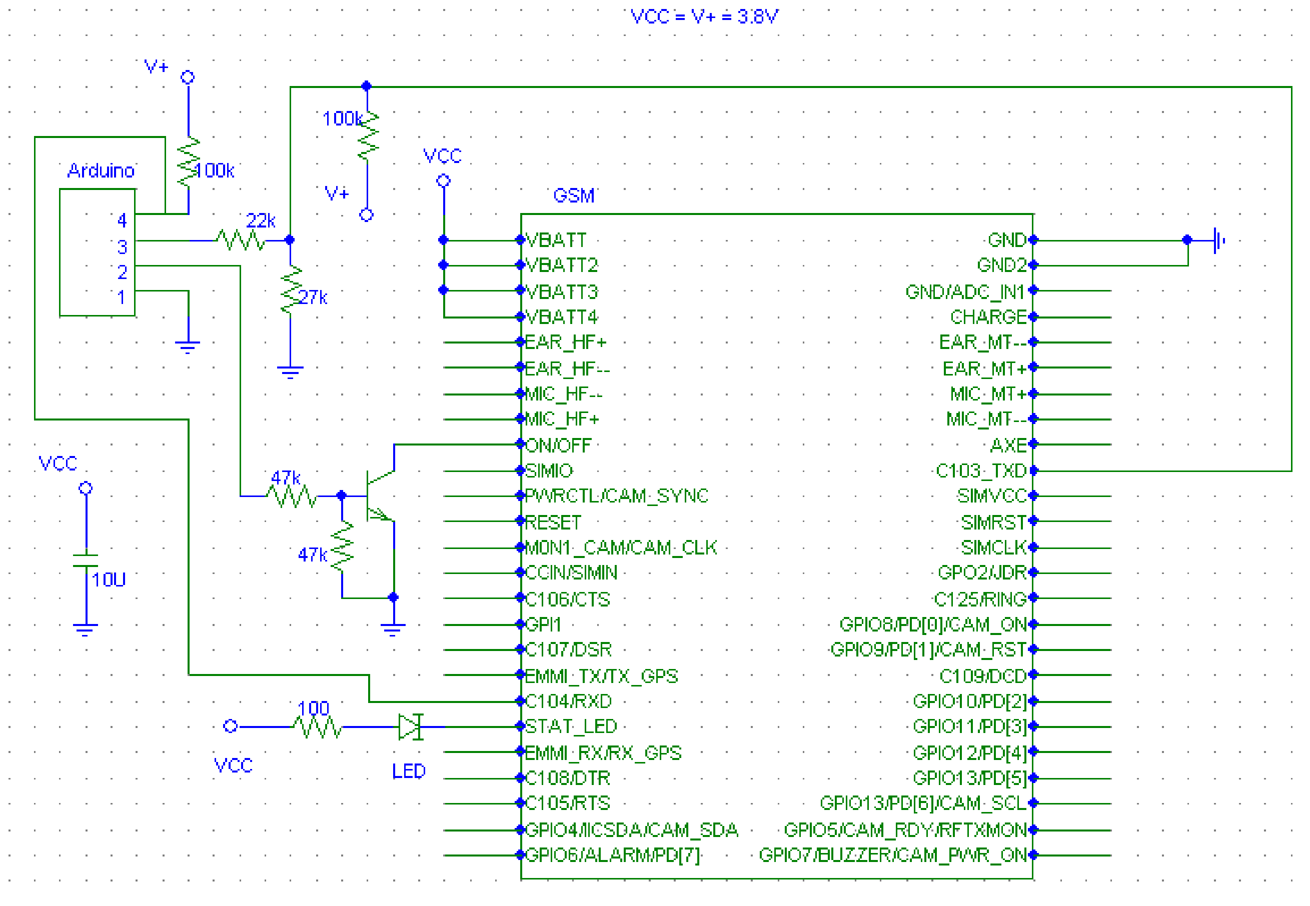

Figure 14: GSM/Arduino circuit diagram

In order to supply the GSM with the necessary 3.8V, a DC/DC converter was used to regulate the voltage down from 6V to 3.8V. The converter used was the PTR08060WVD, made by Texas Instruments. The circuit to operate this component was constructed as shown in figure 15, with Rset=370ohms, which is the necessary resistance to get a 3.8V output [23]. The load in the figure represents the entire GSM circuit shown in figure 14.

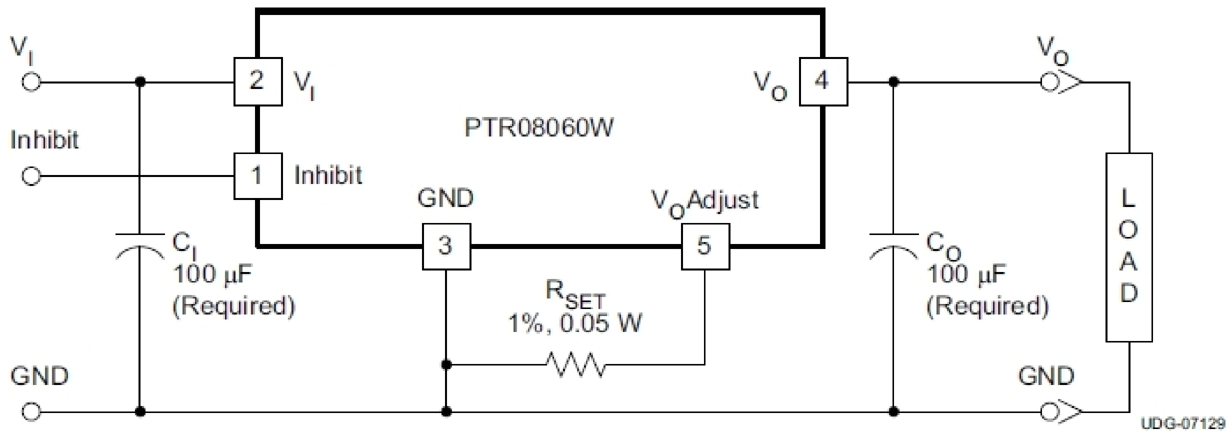

Figure 15: Voltage regulator circuit to supply GSM with 3.8V

Night Sensor

The night sensor circuit is a simple voltage divider with a photoresistor, and it is used so that the Arduino can tell when it is dusk and start trapping at that time. This circuit is diagrammed in figure 16. The photoresistor (Advanced Photonix Inc.: PDV-P9001) was found to have a resistance of about 10 kohms when it was dusk outside; thus, to optimize the resolution of the voltage divider, a 10 kohm resistor was placed in series with the photoresistor. The Arduino reads the voltage across the photoresistor. When it gets dark outside, the resistance increases and the Arduino reads an increased voltage.

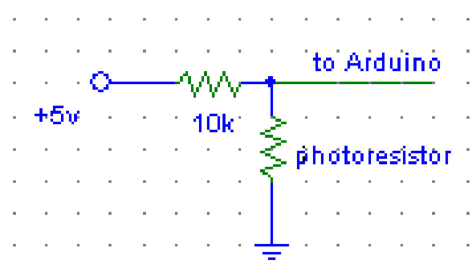

Figure 16: Night sensor circuit diagram.

<u>*Software Development*</u>

In order to control all of the circuit components described above, a program for Arduino had to be developed using the Arduino programming language. Figure 17 shows a flowchart illustrating the general progression of the program.

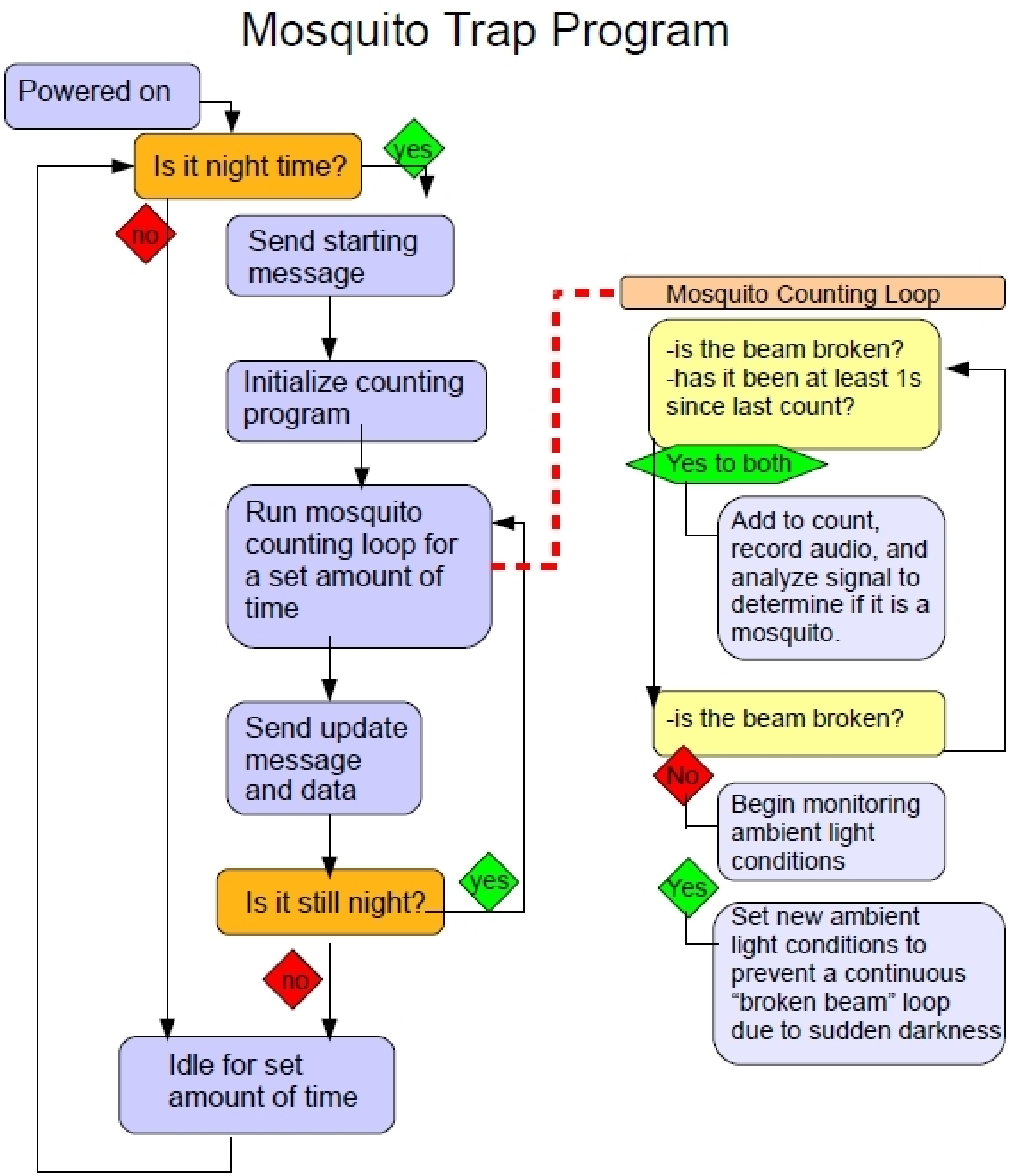

Figure 17: Flowchart of Arduino mosquito trap program

As shown in figure 17, the program has several control loops. The outermost loop checks the night sensor to detect when the sun is setting, which indicates that the trap should begin trapping mosquitoes. Once the night sensor detects dusk, the trap will begin by sending a message via the GSM module to the users, alerting them that the trap has turned on. Following that, both the fan and the laser tripwire are turned on to begin collecting and counting mosquitoes.

While in the counting loop of the trap, the Arduino constantly monitors the laser tripwire, watching for anything that breaks the beam. It does this by detecting changes in light conditions. When the light suddenly gets darker it recognizes this as an insect passing through the beam. The amount of change in light conditions required to set off this response is easily changed to adjust sensitivity. Additionally, the Arduino continuously adapts its value of ambient conditions, so that any drift in the light conditions will not affect the tripwire. To prevent the software from getting caught in a continuously broken beam loop, the trap resets its ambient light conditions every time it detects a broken beam.

A series of events are set off every time the laser beam is broken. The first is that the audio amplifier is powered on and given 10 ms to warm up. Following that, the Arduino records 16 discrete points from the microphone amplifier at a sampling frequency of 8.8 kHz. This data is then used to calculate a 16 point Fast Fourier Transform (FFT). Although a larger FFT would be ideal to improve the differentiation method, because that would allow for the harmonic structures of the mosquito audio signals to be seen, the size of the FFT is limited by the memory available on the Arduino. A simple calculation indicates that a recursive 32 point FFT would take up 1024 bytes of RAM (Arduino has 2 kb of RAM); each float variable takes up 4 bytes, each point would need a real and imaginary part, and the method would use four arrays. The limiting factor, however, is the space in the program memory. It takes 6.7 kb of program memory to run a 16 point FFT; doubling the size of the FFT would nearly double the needed memory, and Arduino only has 14 kb available in program memory, some of which is needed for other parts of the program.

The frequency spectrum obtained from the FFT calculation is then analyzed. It was experimentally determined that the optimal method for determining if a signal came from a mosquito is to take the ratio of frequencies between 0.825 kHz and 1.925 kHz to frequencies between 2.475-4.4 kHz. If this value is above the experimentally determined threshold of 3, then the program identifies the insect as a mosquito. The testing which lead to this method is described later in the testing section.

After the trapping sequence has been running for a set amount of time, the program sends an update message to the users. This message indicates how many bugs have entered in total, how many have been identified as mosquitoes, and the time. The program then checks if it is still night out, and if it is the trap repeats its collection loop, otherwise the trap turns everything off and idles until night.

To improve user-friendliness, several adjustable variables are listed at the top of the program. These include: sensitivity of laser tripwire, threshold of mosquito differentiation method, night darkness threshold, duration between update messages, duration between checking light conditions, and mode of communication (email or text message). A complete copy of the source code is located in the appendix.

Fabrication

Fabrication of the mechanical aspects of the design began once we had determined the final elements in the prototype. We began with construction of the funneling mechanism, which was necessary to ensure the mosquitoes passed through the laser trip wire. The funnel mechanism was made by cutting a standard automotive funnel to fit inside the diameter of the trapping tube (3 inches in diameter). The bottom of the funnel was attached to a CPVC tube 0.5 inches in diameter. Holes were cut in the CPVC to allow for the laser diode, photoresistor, and microphone to rest on the inner wall of the tube to count and differentiate the mosquitoes and other bugs passing through the tube. In order to secure the sensing and differentiating elements, perfboards were placed on either side of the CPVC tube, and the laser diode, photoresistor, and microphone were secured to the perfboards. Another piece of tubing was attached to the perfboards, allowing stabilization for the sensing and differentiating elements within the tube.

Construction of the circuits and electronic aspects was conducted in the bioinstrumentation lab, so that we could test the circuits as we built them. We first put the circuit elements together on a breadboard for testing, and once they worked correctly, we soldered the connections together. The circuit elements were organized by attaching them to perfboard, which held them in place and ensured that the circuit was not shorted. We placed the perfboard with the circuit elements, along with the Arduino and GSM, into the waterproof case. A hole was drilled in the case to allow the wires from the laser trip wire, microphone, and battery to connect to the circuit. Rubber buffers were placed on the case to allow some clearance for the wires to pass out of the trap. Screws were drilled into the side of the funneling tube to allow for stabilization of the case. Wire was run through the case and around the tube, using the screws for stabilization, to secure the case. Once all the elements were in place, the final wiring was soldered and testing was conducted.

Testing

Laser Trip Wire

The laser trip wire was tested in the lab by interrupting the beam with a thin wire. Once it was working, it was brought to the mosquito lab to be tested with real mosquitoes. The mosquitoes were drawn into a tube and blown across the beam. It was difficult to tell if the mosquito had crossed the beam or not, but from careful observation and adjustment of the sensitivity of the counting program on Arduino, it was determined that the system had a very high success rate. It was experimentally determined that the optimal sensitivity for the laser trip wire was 1.06. This value is defined as the coefficient of deviation from ambient conditions required for the Arduino to count it as a positive trip. Sensitivities lower than 1.04 resulted in false positives, whereas sensitivities greater than 1.08 did not always sense the mosquitoes. In the final design, the sensitivity was lowered to 1.035 because this was the lowest value which did not produce false positives in the more stable light conditions on the inside of the PVC tube.

GSM

The GSM testing took up a lot of time because we were given a faulty device. We had it hooked up to Arduino to try and get it to send a test message to one of our phones. We initially thought the problem lay in the code in the Arduino program, so we went through line by line rewriting the code to try and correct the problem. We used a multimeter to see where the voltage was being carried, and eventually it was determined that Arduino was correctly sending the signal to send the message, but the GSM module was not receiving it. We obtained a new GSM module, and once we get it hooked up, it sent the text messages that we wanted it to. Additionally, we tested sending text messages to emails, which was successful.

Microphone

To test the microphone circuit to see if it worked correctly, the output was connected to an oscilloscope and someone would whistle into it or a 2 kHz tone was played from the computer to see if the frequency could be picked up. Once the circuit was working consistently, the Fast Fourier Transform (FFT) was taken of the sound when the 2 kHz sound was played to see if the coding and method for the frequency determination through Arduino was accurate. The graphs were analyzed to determine which frequency they had picked up, and they consistently read a value of 2 kHz. When the circuit was performing properly in the lab, we took the microphone to the mosquito lab to measure the frequency patterns of mosquitoes that had been placed into a jar along with the microphone. We measured the frequencies of three different species, *Anopheles stephensi*, *Culex pipens*, and *Aedes aegypti*. A lot of the signals had interfering noise from outside sources, such as equipment within the lab room, which resulted in noisy graphs. Additionally, the mosquitoes were most often standing still, so the audio signals obtained had no trace of the sound of their wings beating. However, several quality samples were collected, which had little noise and distinct patterns of wing beats.

With the limited processing power of Arduino, which can only do a 16-point FFT, it was difficult to determine exactly what species was being heard. It would require approximately a 128-point FFT in order to accurately determine the species of the mosquito, because it would require visualizing the harmonic patterns produced by the wing beats. With the signals we obtained from actual mosquitoes, we examined 65 16-point FFT's to search for a reliable algorithm to differentiate mosquitoes from other bugs. We also examined 49 16-point FFT's from negative controls, including: a recording of the fan used in the trap without any other noise, and online recordings of crickets, a swarm of midges, and a variety of night insects. From this testing, it was determined that the ideal algorithm was to compare the ratio of frequencies between 0.825 kHz and 1.925 kHz to those between 2.475-4.4 kHz. A high threshold was chosen for the value of this ratio, in order to prevent as many false positives as possible. Based on the 114 samples tested, this method had a specificity 98% and sensitivity of 23%.

Light Sensor

The light sensor turns the trap on at night and turns it off in the morning by using the readings of a photoresistor placed on the outside of the waterproof box. In order to tell Arduino what resistance values correlated with morning and night, data was collected every five minutes for several hours, spanning through the afternoon and into the evening. The value of the photoresistor in direct light was 0.3 kohms and at dusk it was 10 kohms, so the threshold resistance was chosen to be 4.6 kohms.

Entire Final Design

The battery, SMS messaging system, and night sensor were tested in a final test of the trap. The battery lasted somewhere between 8 and 10 hours, determined by the number of text messages sent. The texts were coded to be sent every two hours, although this is going to be changed to whatever frequency the researchers desire, most likely one hour. The texts included information about the number of mosquitoes detected and the number of other bugs that entered the trap.

The entire trap has yet to be tested outdoors, but we do not have the resources to do this. In order to complete testing, the system would be left out for the night with the rest of the gravid trap components including the container of stink water and the mesh collecting bag. The results, determined in the current manner of manually analyzing the insects collected, would be compared to the reports from Arduino. It is important that the count of mosquitoes that have entered the trap correlates to the actual number of mosquitoes in the trap, so that the information can show the entomolgists trends of mosquito populations over the course of the season.

Mechanical

In order to make a secure final product, the mechanical aspects integrated into the mosquito trap needed to allow for the least possible intrusion upon how the mosquito trap functions. We were able to accomplish this with elaborate testing of all of the mechanical pieces.

First, after designing the funnel and tube that connects to it, it was necessary to make sure that the air flow through the trap was still adequate, as the funnel blocked some of the flow. The funnel was an integral piece of the project, as it forced mosquitoes into a tube which housed the laser trip wire, allowing each mosquito to be individually counted and added to a running tally. The testing of this device was accomplished by taking simulation mosquito models, “phantoms,” which consisted of crumpled pieces of kimwipe paper, and making sure the fan was powerful enough to propel them through the trap. The reason kimwipe was chosen as a simulation mosquito was because it accurately depicted their weight and was made into pieces that were about the size of a mosquito. We discovered that the phantoms were not being sucked into the trap because the funnel blocked airflow through the trap. We cut the funnel and replaced the walls with mesh, in order to allow the air to flow through the trap. We then discovered that the phantoms were still not passing through the funnel, and were only bouncing around in a helical pattern. This was because the fan produced a helical air flow pattern, and the phantoms were not traveling vertically through the tube. We implemented a rectangular piece of perfboard and secured it below the funnel to change the airflow to a path parallel to the walls of the tube, which caused the phantoms to be pushed straight through the funnel to the other side.

Once the funnel proved to be effective, the perfboard, funnel, and tubing containing the laser trip wire and microphone were secured to the inside of the tube using hot glue. Upon three series of tests, we achieved an 85 percent completion rate. In the first test, 17 out of 20 simulation mosquitoes made it through the trap, the second gave 16 out of 20 and the third gave 18 out of 20. This testing proved that the airflow was sufficient with the funnel inside of the trap and that there were minimal rough edges on the funnel that mosquitoes could get caught on. Also figured into this test, real mosquitoes, should they get stuck, will instinctively push themselves through the hole in order to get out of the wind. This is virtually the only way that they can go, as they will not be strong enough to fly against the wind. Given this information, we are confident that the airflow is sufficient and mosquitoes will still be forced into the trap and propelled through it to the collection bag.

One further series of tests that were performed were to test the strength of the wire connecting the waterproof box to the tube portion of the trap. The final design product is meant to be durable and long-lived, and the connection between the box and tube needed to be strong in order to realize this goal. In connecting the box, thick black wire was used. Before deciding upon this, the wire was put through a series of tests, which determined its load capacity to approximately 40 pounds. When the wire was used to connect the tube and box, it was doubled up, which means that it would take approximately 80 pounds of force to break the connection. This is significantly more force than it will ever see, which will result in a durable product. Because of the testing process, the final product is mechanically sound.

Future Work

In the future, there are still improvements that can be made to the mosquito trap in order to make it both more efficient and more effective. Despite the fact that upgrades were developed for the trap in the key areas specified by the clients, there were some possible minor improvements that were not addressed. These improvements include integrating a temperature sensor within the trap and implementing a device that kills mosquitoes as they enter the trap. Additionally, if the clients wanted to implement this design with the light traps, further improvements could focus on replacing the incandescent bulb with an LED and standardizing the release of carbon dioxide. These design aspects were discussed with our clients and ultimately they were decided to be secondary goals. In the future, work should be done in these areas in order to further enhance the mosquito trap.

Of the areas that we did work on, several could use slight improvement. As it is right now, the mosquito trap can only calculate a 16-point fast-Fourier transform. In order to be more accurate in identifying mosquitoes, it would be beneficial to provide the means to conduct a larger transform. This could be done by integrating a stand alone FFT processor to make all the calculations separate from the Arduino. Along with this improvement, a better method for capturing the sound from individual mosquitoes could be developed. As of right now, there can be a lot of background noise, which disrupts the microphone and has the potential to slightly alter the results. One additional area of improvement is in the category of physical construction. While our final product is durable and rugged, many of our components could be improved. The final design we have created will work well as a prototype, but if it was ever to be used on a large scale it would be beneficial to engineer parts specially made for the project. As an example to this, our waterproof housing for the electrical components of the project is fastened to the mosquito trap tube with wire. While the wire holds well, a production model would replace it with an alternative more suited for the job. All of these things being said, the final product is able to completely function as it is right now and has the potential to be implemented in the field.

Though our client is David Van Sickle, the final product will be delivered to Susan Paskewitz and Patrick Irwin, entomologists. The finalized trap has the potential to be used in the field for further testing. Once these test runs have been completed, we will have a further indication as to the future work that needs to occur before this trap can become widely used. At this time, we would like to thank David Van Sickle, Chris Brace, Susan Paskewitz, and Patrick

Irwin, Tim Balgemann, Amit Nimunkar and Tim Theisen for helping us through the entire process. The final project will be delivered shortly after the final presentation on May 7.

References:

1. Gorman, Christine. (2004). Death by mosquito. *Time Magazine*, Retrieved from http://www.time.com/time/magazine/article/0,9171,1101040726-665039,00.html

2. Mosquito solutions. (2008). *Mosquito Facts*, Retrieved from http://www.control-mosquitoes.com/

3. Milda: mosquito light traps. (2008). *Milda Light Traps*, Retrieved from www.milda-cevre.com/en/diger-urunlerimiz

4. Guidance for use of the cdc gravid trap. (2005). *Gravid Trap Instructions*, Retrieved from http://chppm-www.apgea.army.mil/ento/westnile/main/TabC_GuidelinesGarvidTrap

5. Norfolk county mosquito surveillance program. (2008). *Mosquito Monitoring Program*, Retrieved from www.massnrc.org/ncmcp/surveillance.htm.

6. Mosquito monitoring and surveillance. (2008). *State of Connecticut*, Retrieved from http://www.ct.gov/mosquito/cwp/view.asp?a=3486&Q=414712&mosquitoNav=%7C

7. AMCA traps. (2005). *AMCA*, Retrieved from http://www.mosquito.org/mosquito-information/traps.aspx

8. Price Sheet and Order Form. (2010). John W. Hock Company. http://www.johnwhock.com/buy/index.htm

9. Arduino Diecimila. (2010). Arduino. http://Arduino.cc/en/Main/ArduinoBoardDiecimila

10. Plastic Enclosures and Electrical Connector Enclosures. (2010). PacTec. http://www.pactecenclosures.com/

11. Gardber, J. P. and Demattia, D. N. (2006). *Light extinction based non-destructive flying insect detector.* U.S. Patent 7071829.

12. Features and Specifications of Laser Tripwires. (2010). Tripwire Enterprises. http://www.tripwire.com/it-compliance-products/te/specifications/

13. Laser Diode. (2010). Digikey Corp. http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=67-1500-ND

14. MP Motion Sensors. (2009). Panasonic. http://www.mouser.com/panasonicnapion/

15. GSM Module. (2010). Nenomart. http://www.nenomart.com/catalog/product_info.php?products_id=171&osCsid=5amb7out9vs98mtrpirepv6v27

16. Embedded Systems. (2009). Amtel corp. http://embedded-system.net/

17. Raman, D. R., Gerhardt, R. R., & Wilkerson, J. B. (2007). Detecting Insect Flight Sounds in the Field:
Implications for Acoustical Counting of Mosquitoes. *American Society of Agricultural and Biological Engineers, 50*[4], 1481-1485.

18. Shaw, J. A., Repasky, K. S., Carlsten, J. L., Spangler, L. H., & Hoffman, D. S. (2009). *Optical detection of oscillating targets using modulation of scattered laser light.* U.S. Patent 7,511,624.

19. WM-55D103 Electret Condenser Microphone. (2010). Digikey. http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=P9897-ND

20. Using a MOSFET as a Switch. (2010). Brunning Software. http://brunningsoftware.co.uk/FET.htm

21. Interfacing Arduino with a Telit GM862. (2009). Tinkerlog. http://tinkerlog.com/2009/05/15/interfacing-Arduino-with-a-telit-gm862/

22. GM862 Family Hardware User Guide. (2009). Telit Wireless Solutions. http://www.gm862.com/en/products/gsm-gprs.php?p_id=12&p_ac=show&p=4

23. PTR08060WVD Datasheet. (2009). Digikey. http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=296-22673-5-ND

Appendix

<u>*Product Design Specifications*</u>

February 12, 2010

Function

This project is designed to apply a variety of electronic technologies to current light trap models to improve their utility and ease of use and their potential to generate valuable, timely data for public health. In particular, we plan to integrate a variety of sensors to collect data on the local environment and on the trapped mosquitoes, and provide a means to remotely monitor and operate the device.

Client Requirements

- The mosquito trap should be durable and rugged
- Withstand weather and exposure
- Reliably capture and count an accurate and valid sample of mosquitoes of interest
- Mechanically limit the number of other insects counted and captured
- Must not diminish length of trap operation
- Relatively simple to operate
- Could be designed as add-on unit to current traps (rather than an entirely new trap)

Design Requirements

1. <u>Physical and Operational Characteristics</u>
    a. *Accuracy and Reliability:* The device must produce meaningful mosquito population data, so it should accurately and reliably measure the quantity of mosquitoes entering the trap.
    b. *Life in Service:* Must last at least overnight, preferably the device should last considerably longer to minimize human effort.
    c. *Shelf-life:* The shelf-life should be nearly indefinite so that it can be used over many seasons, although certain components may need to be recharged/replaced (e.g. batteries, CO_2 cartridges, etc.)
    d. *Operating Environment:* The device will be used in an outdoor environment; it should be able to withstand rain, wind, dirt, dust, and other common outdoor conditions. It will have minor protection from the environment from the current trap design which has a circular "roof".
    e. *Safety:* The device must be not put the user in any danger. Potential sources for danger include: electrical components, mechanical aspects, and pressurized gases.
    f. *Ergonomics:* The device must be designed to allow the user to easily set up the trap for data collection and access the mosquitoes collected
    g. *Size:* The device should be small enough to facilitate easy transport to a mosquito collection site, but is otherwise unlimited.
    h. *Weight:* The device should be light enough to facilitate easy transport to mosquito collection site, but is otherwise unlimited.

2. <u>Production Characteristics</u>

a. *Quantity:* One prototype should be constructed.
b. *Target Production Cost:* Up to $1000 for development, but the final design should cost significantly less The device will potentially save much time in setting up and collecting data, so it can be more costly than other models.

3. <u>Miscellaneous</u>
   a. *Standards and Specifications:* The device must produce meaningful data as compared other mosquito traps.
   b. *Customer:* The primary customers are our clients; their main concern is to produce a mosquito trap that requires less human effort to operate, while maintaining a valid collection of data.
   c. *Competition:* There are many mosquito traps available, but few, if any, are capable of real-time data acquisition and remote operation.
   d. *Priorities:* Due to a wide variety of potential improvements, the following priorities have been established to guide the process:
      1) Establishing a method of sensing or counting mosquitoes
      2) Integrating this method with a means of remote control and access of data
      3) Integrating a means of differentiating mosquitoes from other insects or, ideally, speciating the mosquitoes.
      4) Improving upon other aspects of the trap, such as mosquito attractants, power sources, a variety of environmental sensors, a means of killing mosquitoes upon entry, or many more.

<u>Mosquito trap program source code</u>

```
/*
*  ----MosquitoTrap----
*
*  Purpose: to operate functions of a mosquito trap,
*    including toggling normal trap features (fan, light, etc)
*    on and off, monitoring the mosquito detection sensor,
*    acquiring audio signals, and communicating with GSM
*
*  Developers: BME 201 Mosquito Trap Team Spring 2010
*     Jeff Theisen, John McGuire, Courtney Krueger, and Ryan Nessman
*
*  Version: 1.0
*  Date: May 1, 2010
*/
//pin locations
int ledPin = 13;          //LED connected to 13, used for testing purposes to show detection
int laserPin = 8;         //laser for tripwire
int laser_resistorPin = 7; //photoresistor of tripwire
int laser_sensorIn = 0;   //analog input pin for sensor photoresistor
int audioToggle = 5;      //mosfet to toggle audio amplifier on/ff
int audioSensor = 1;      //analog input pin for microphone
int fanToggle = 6;        //mosfet to toggle fan/light on and off
int nightSensorOut = 4;   //Vout for night time voltage divider
int nightSensorIn = 2;    //analog input to read voltage divider of night time sensor
int GSMonOff = 2;         //pin to turn GSM modulne on/off
int regulator = 3;        //pin to toggle GSM's voltage regulator on/off
//RX and TX (pin 0 and 1) are also connected, but require no declaration
```

```
//other variables to change for optimization
boolean send_email = true;          //true = email message, false = text message, will not send both
char phone[] = "1608xxxxxxx";       //INTERNATIONAL format - phone to send texts too
char email[] = "email@wisc.edu"; //desired destination for output
double sensitivity = 1.035;         //decrease towards 1.0 to increase sensitivity of sensor at detecting a bug entering (recommended 1.02-1.06)
float threshold = 3;                //arbitrarily selected value for FFT differentiation method (higher = more sensitive)
int nightThreshold = 400;           //higher value waits for darker conditions (0-1023, 0-30=direct sunlight, 900-1023=night)
int idleDelay = 60000;              //MILLISECOND delay between checking if it's night (ms)
long updateDelay = 3600000;         //MILLISECOND delay between sending update messages

//variables used by program (recommended not to change these)
int rawCount = 0;         //counts number of times insects are detected
int positiveCount = 0;    //number of counts likely most likely mosquitoes
double ambientLight;      //a baseline value for comparison in laser tripwire
boolean nightTime = false; //program will being sensing during night
float temp;               //placeholder for numerous calculations
float data[16];           //audio input
float output[16];         //fft output array
boolean beamBroken = false;      //used in laser tripwire
boolean recentlyTripped = false; //used in laser tripwire
long timer;               //keeps track of how long tripwire has been running
boolean starting = true;  //needed to send init message just once

void setup(){
  pinMode(GSMonOff, OUTPUT);
  pinMode(regulator, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(laserPin, OUTPUT);
  pinMode(laser_resistorPin, OUTPUT);
  pinMode(audioToggle,OUTPUT);
  pinMode(fanToggle, OUTPUT);
  pinMode(nightSensorOut, OUTPUT);
  Serial.begin(4800);

  digitalWrite(regulator, HIGH); //inhibits regulator (keeps it off)

  for(int i=0;i<10;i++){//show that the trap is connected
    digitalWrite(laserPin,HIGH);
    delay(500);
    digitalWrite(laserPin,LOW);
    delay(500);
  }

  //IS IT NIGHT TIME?
  checkNight();

}


void loop(){

  while(nightTime){

    //SEND INITIALIZING MESSAGE
    if (starting){
      initMessage();
      starting=false;
    }

    //INITIALIZE COUNTING PROGRAM

    digitalWrite(fanToggle, HIGH);    //turn on fan and light
    digitalWrite(laserPin, HIGH);     //turning on the
    digitalWrite(laser_resistorPin, HIGH);   //sensor

    ambientLight = analogRead(laser_sensorIn);  //get a measure of ambient
    for (int i =0;i<20;i++){             //light for future comparison
      delay(500);
      ambientLight = ambientLight*0.95 + 0.05*(analogRead(laser_sensorIn));
    }//end ambient light measuring
```

```
    //RUN SENSING/COUNTING PROGRAM
    beamBroken = false;       //has the beam been broken
    recentlyTripped = false;  //needed to avoid double counts
    timer = millis();         //keep track of time
    timer += updateDelay;

    //loops until it's time to send data
    while((millis())<(timer)){

      if(analogRead(laser_sensorIn) > (ambientLight*sensitivity)){
        beamBroken=true;
      }else{
        beamBroken=false;
      }

      if(beamBroken && !recentlyTripped){ //if beam is broken and hasn't been recently
        rawCount++;
        recentlyTripped = true;

        //RECORD/ANALYZE AUDIO SIGNAL
        audioProcessing();
        delay(500);//avoid double counting
      }//end broken beam sequence

      if (!beamBroken){
        recentlyTripped=false;
        ambientLight = ambientLight*0.9999 + 0.0001*(analogRead(laser_sensorIn));//adjust ambient light measurement

      }else{//in case of sudden darkness, this will prevent from getting stuck in a beambroken/recentlytripped loop
        ambientLight = analogRead(laser_sensorIn);
      }

    }//pause counting program to send message

    //IS IT STILL NIGHT?
    checkNight();

    //SEND UPDATE MESSAGE
    updateMessage();


  }//night time is over, idle for the day
  digitalWrite(laserPin, LOW);            //turning off the
  digitalWrite(laser_resistorPin, LOW);   //sensor
  digitalWrite(fanToggle, LOW);           //turn off fan
  rawCount=0;                             //reset counts
  positiveCount=0;
  starting=true;

  //IDLE PROCESS
  delay(idleDelay);
  checkNight();


}


void initMessage(){
  digitalWrite(regulator, LOW); //allow 3.8V to go into GSM
  delay(5000);
  gsmOnOff();
  gsmBoot();

  //establish baud rate
  Serial.print("AT+IPR=4800\r\n");
  delay(300);

  //set up SMS message type
```

```
  Serial.print("AT+CMGF=1\r\n");
  delay(300);

  //send SMS message
  if(send_email){
    Serial.print("AT+CMGS=+121\r\n");
    delay(300);
    Serial.print(email);
    delay(300);
    Serial.print(" ");
  }else{
    Serial.print("AT+CMGS=+");
    delay(300);
    Serial.print(phone);
    delay(300);
    Serial.print("\r\n");
  }

  delay(300);
  Serial.print("Mosquito Trap has initialized collection sequence.");
  delay(300);
  Serial.print(0x1A,BYTE);
  delay(30000); //wait for SMS to  send

  gsmOnOff();  //turn gsm off
  digitalWrite(regulator, HIGH);//turn of regulator to conserve power

}

void updateMessage(){
  digitalWrite(regulator, LOW); //allow 3.8V to go into GSM
  delay(5000);
  gsmOnOff();
  gsmBoot();

  //establish baud rate
  Serial.print("AT+IPR=4800\r\n");
  delay(300);

  //set up SMS message type
  Serial.print("AT+CMGF=1\r\n");
  delay(300);

  //send SMS message
  if(send_email){
    Serial.print("AT+CMGS=+121\r\n");
    delay(300);
    Serial.print(email);
    delay(300);
    Serial.print(" ");
  }else{
    Serial.print("AT+CMGS=+");
    delay(300);
    Serial.print(phone);
    delay(300);
    Serial.print("\r\n");
  }

  delay(300);
  Serial.print("Mosquito Trap Update\n\n");
  delay(300);
  Serial.print("Total nightly progress...\n");
  delay(300);
  Serial.print("Bugs collected: ");
  delay(300);
  Serial.print(rawCount);
  delay(300);
  Serial.print("\nNumber identified as mosquitoes: ");
  delay(300);
  Serial.print(positiveCount);
```

```
  delay(300);

  if (!nightTime){
    Serial.print("\n\nDaytime detected, idling.");
    delay(300);
  }

  Serial.print(0x1A,BYTE);
  delay(30000); //wait for SMS to send

  gsmOnOff();  //turn gsmOff
  digitalWrite(regulator, HIGH);//turn of regulator to conserve power

}

void gsmOnOff(){
  digitalWrite(GSMonOff, HIGH);
  delay(2000);
  digitalWrite(GSMonOff, LOW);
  delay(1000);
}

void gsmBoot(){//allow GSM a minute to find network, send messages to keep it from idling)
  for(int i=1;i<60;i++){
    delay(1000);
    Serial.print("AT");
  }
  delay(1000);
}


void checkNight(){
  digitalWrite(nightSensorOut, HIGH);
  delay(10);
  if (analogRead(nightSensorIn) > nightThreshold){
     nightTime=true;
  }else{
     nightTime=false;
  }
  digitalWrite(nightSensorOut, LOW);
}

void audioProcessing(){
  digitalWrite(audioToggle, HIGH); //turn on amplifier
  delay (10);      // let amplifier warm up

  //collect data
  for(int i=0;i<16;i++){
      data[i]=analogRead(audioSensor);
  }
  digitalWrite(audioToggle, LOW);  //turn off amplifier

  //adjust sample
  temp=0;//average
  for(int i=0;i<16;i++){
    temp+=data[i];
   }
   temp/=16;
   for(int i=0;i<16;i++){
     data[i]-=temp;
   }
   //fft
   fft(data,output);
   //absolute values
   for(int i=0;i<16;i++){
      if(output[i]<0) output[i]*=-1;
   }

   //likelihood of mosquito. 3+4/6-9
   temp = (output[2]+output[3])/(output[5]+output[6]+output[7]+output[8]);
```

```
    if(temp>threshold){
      positiveCount++;
    }

}

void fft(float input[16], float output[16]){
  float temp, out0, out1, out2, out3, out4, out5, out6, out7, out8;
  float out9,out10,out11,out12,out13,out14,out15;
  float pi = 3.141592653589793;

  out0=input[0]+input[8]; /* output[0 through 7] is the data that we */
  out1=input[1]+input[9]; /* take the 8 point real FFT of. */
  out2=input[2]+input[10];
  out3=input[3]+input[11];
  out4=input[4]+input[12];
  out5=input[5]+input[13];
  out6=input[6]+input[14];
  out7=input[7]+input[15];

  out8=input[0]-input[8];   /* inputs 8,9,10,11 are */
  out9=input[1]-input[9];   /* the Real part of the */
  out10=input[2]-input[10]; /* 4 point Complex FFT inputs.*/
  out11=input[3]-input[11];
  out12=input[12]-input[4]; /* outputs 12,13,14,15 are */
  out13=input[13]-input[5]; /* the Imaginary pars of  */
  out14=input[14]-input[6]; /* the 4 point Complex FFT inputs.*/
  out15=input[15]-input[7];

  /*First we do the "twiddle factor" multiplies for the 4 point CFFT */
  /*Note that we use the following handy trick for doing a complex */
  /*multiply:  (e+jf)=(a+jb)*(c+jd) */
  /*   e=(a-b)*d + a*(c-d)   and    f=(a-b)*d + b*(c+d)  */

  temp=(out13-out9)*(sin(2*pi/16));
  out9=out9*(cos(2*pi/16)+sin(2*pi/16))+temp;
  out13=out13*(cos(2*pi/16)-sin(2*pi/16))+temp;

  out14*=(sin(4*pi/16));
  out10*=(sin(4*pi/16));
  out14=out14-out10;
  out10=out14+out10+out10;

  temp=(out15-out11)*(sin(6*pi/16));
  out11=out11*(cos(6*pi/16)+sin(6*pi/16))+temp;
  out15=out15*(cos(6*pi/16)-sin(6*pi/16))+temp;

  /* The following are the first set of two point butterfiles */
  /* for the 4 point CFFT */

  out8+=out10;
  out10=out8-out10-out10;

  out12+=out14;
  out14=out12-out14-out14;

  out9+=out11;
  out11=out9-out11-out11;

  out13+=out15;
  out15=out13-out15-out15;

  /*The followin are the final set of two point butterflies */
  output[1]=out8+out9;
  output[7]=out8-out9;

  output[9]=out12+out13;
  output[15]=out13-out12;

  output[5]=out10+out15;        /* implicit multiplies by */
```

```
  output[13]=out14-out11;        /* a twiddle factor of -j */
  output[3]=out10-out15;  /* implicit multiplies by */
  output[11]=-out14-out11;  /* a twiddle factor of -j */


  /* What follows is the 8-point FFT of points output[0-7] */
  /* This 8-point FFT is basically a Decimation in Frequency FFT */
  /* where we take advantage of the fact that the initial data is real*/

  /* First set of 2-point butterflies */

  out0=out0+out4;
  out4=out0-out4-out4;
  out1=out1+out5;
  out5=out1-out5-out5;
  out2+=out6;
  out6=out2-out6-out6;
  out3+=out7;
  out7=out3-out7-out7;

  /* Computations to find X[0], X[4], X[6] */

  output[0]=out0+out2;
  output[4]=out0-out2;
  out1+=out3;
  output[12]=out3+out3-out1;

  output[0]+=out1;  /* Real Part of X[0] */
  output[8]=output[0]-out1-out1; /*Real Part of X[4] */
  /* out2 = Real Part of X[6] */
  /* out3 = Imag Part of X[6] */

  /* Computations to find X[5], X[7] */

  out5*=sin(4*pi/16);
  out7*=sin(4*pi/16);
  out5=out5-out7;
  out7=out5+out7+out7;

  output[14]=out6-out7; /* Imag Part of X[5] */
  output[2]=out5+out4; /* Real Part of X[7] */
  output[6]=out4-out5; /*Real Part of X[5] */
  output[10]=-out7-out6; /* Imag Part of X[7] */

}
```