SMART TRAP

ECE 445

TEAM 63

Jonathan Drougas
Christian Morales
Xiaowei Yuan

TA: Bonhyun Ku

MAY 2021

Abstract:

The following report explains the research and development of our ECE 445 project, Smart Trap. Smart Trap ensures that domestic animals, infants, or otherwise unintended creatures cannot become stuck in a trap's cage. The device complements existing pressure plate style animal traps. Design strategy, the troubleshooting process, associated costs, and further developments are explained in the body of the report.

Contents

1. Introduction

Above: A Raccoon Inside a Pressure-Plate Trap

The goal of this project is to provide an addon for existing animal traps that would be able to capture targeted creatures like raccoons while leaving untargeted creatures like cats, dogs, and infants uncaptured. This would cut down the cost on the consumer from having to purchase a proprietary trap if they already own one, while at the same time cutting the cost of having to purchase a cage for each of the products being sold. The Smart Trap appeals to suburban homeowners, outdoorsmen, and conservationists alike.

1.1. Problem Statement

Creatures like raccoons have been a problem plaguing households. [1] While there exist conventional pressure plate activated traps on the market, these traps are not able to distinguish between creatures like raccoons and pets like dogs and cats. This means that the trap could potentially capture one of these pets instead of the actual creature. In the worst case, a small child could become entrapped.

1.2. Solution

The solution to this problem is to design an addon which would detect whether the creature that has entered the trap is a targeted creature. If it is not a targeted creature, then the creature can enter the trap without being captured by the trap. If it is a targeted creature, then the creature would be captured by the trap.

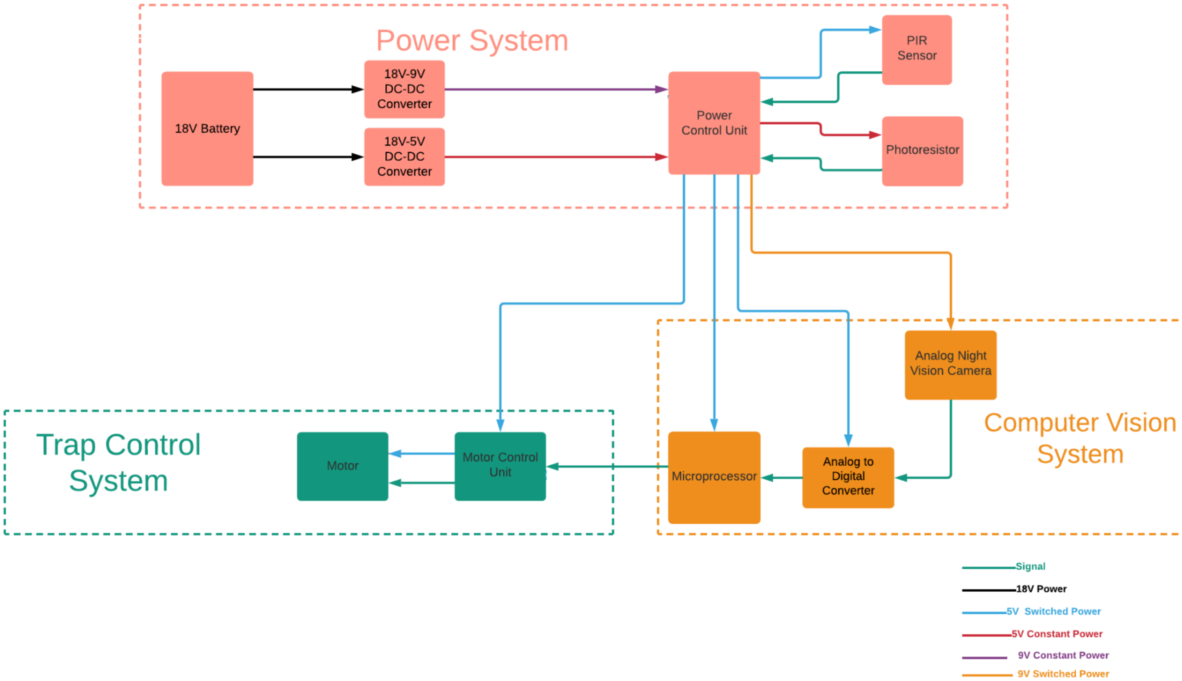
1.3. Wholistic Implementation

A pressure plate style trap operates by releasing a cage door when an animal's weight depresses the pressure plate. Our team chose to design a device which defeats a pressure plate style trap by default. Once the targeted animal is detected, the trap is armed and can function as it normally would. We accomplished this by using a servo motor to press on the underside of the trap's pressure plate. The servo holds the pressure plate in place so that no creature can become trapped in the cage. When the target animal is detected by our device, the servo no longer blocks the pressure plate and the trap functions as it normally would. This simple approach of defeating the trap saves on the need to design a mechanism which controls the trap's door. Therefore, cost is greatly reduced using this method.

2. Design

This section elaborates on the technical design considerations of our project. The final revisions of hardware and software are explained as well as how those designs were modified throughout the span of the semester.

2.1. Block Diagram



Above: Smart Trap Block Diagram

2.1.1. Block Diagram: Design Overview

The smart trap design is broken down into three major components in order to make the product work. The power system is used to control power sent to the different components in the device, while also minimizing the power used. A computer vision system handles the detection and identification of the creature entering the trap, while also signaling to the trap control system. Lastly, the trap control system rotates the servo when a targeted creature has been spotted by the trap.

2.1.2. Block Diagram: Changes and Updates

The main change made between the initial block diagram from the design document and the revised block diagram shown above is the dissolution of the sensing system, which contained the PIR sensor, the photoresistor and the analog night vision camera. The PIR sensor and the photoresistor have been incorporated into the power system, while the analog night vision camera has been incorporated into the computer vision system.

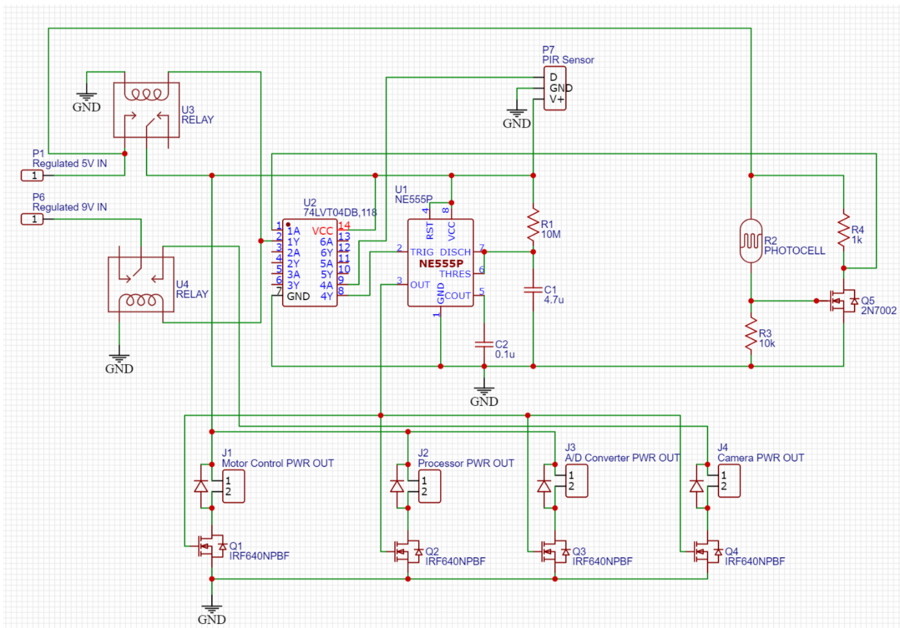
2.2. Power Control Module

The power control module consists of several components meant to supply and moderate power to each component of the Smart Trap. For the final design, the parts of the power system include an 18V tool battery, two DC-DC voltage converters, a passive infrared sensor, and an analog circuit which controls power.

2.2.1. Power Control Module: Purpose

The purpose of the power control module is to conserve energy. Optimally, the required battery capacity should be minimized. Conserving energy also helps to achieve our goal of having greater than twelve hours of battery life. A simple solution to this problem is to use a larger battery, but this is costly and increases recharge time. This also increases the physical space needed for our device.

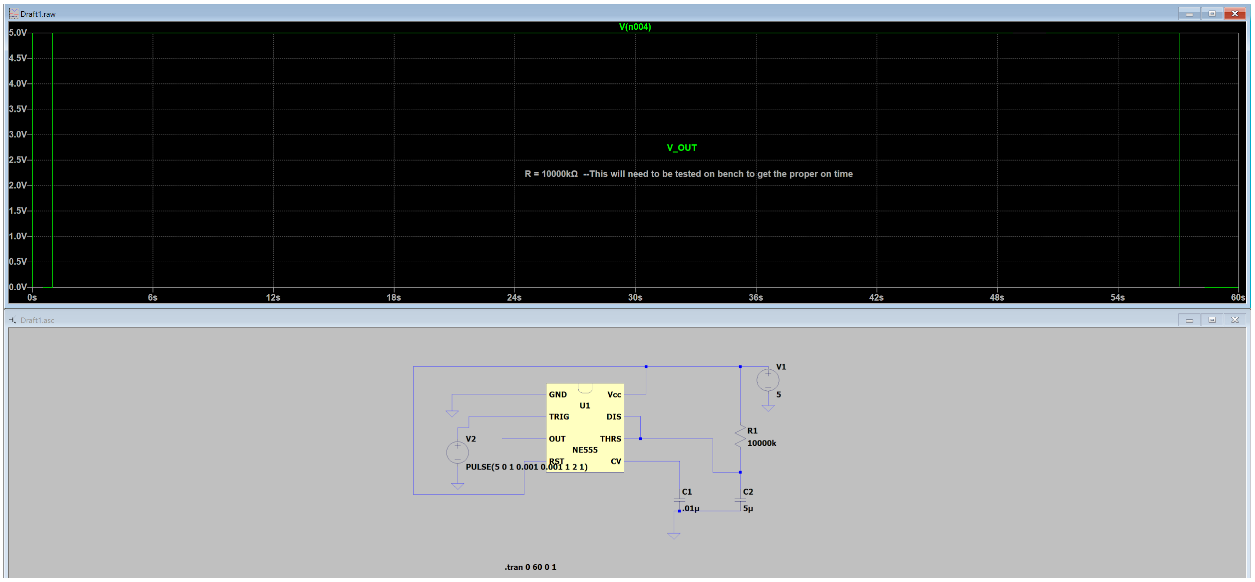
2.2.2. Power Control Module: Schematic



Above: Power Control Module Circuit Schematic

2.2.3. Power Control Module: Implementation

There are two main sub-circuits in the power control module. The first sub-circuit is responsible for switching power to the A/D converter, the camera, and the motor control circuit. In our final design, this sub-circuit should also tell the microprocessor to enter a low power state. This functionality is accomplished by using a passive infrared sensor to detect the presence of animals. When an animal is detected, the PIR sensor emits a pulse. The sub-circuit extends this pulse to roughly one minute. This is done using a 555 timer circuit seen in the center of the schematic above. The 555 timer was researched independently and some circuit design influence was taken. Designs were modified and tested in simulation using LTSpice. The RC time constant created by R1 and C1 set the pulse length. [2] The output on pin three of the timer is then used to switch power to the various components using NMOS devices. A diode is placed across the output for each switched device in case there is any inductance in the load in order to protect the MOSFETs from back EMF. [3] The simulated output pulse can be seen below.



Above: PIR Pulse Extender Circuit Schematic and Simulation Result

The second sub-circuit is responsible for shutting power off to the system during daylight. Because the trap is designed to capture nocturnal creatures, switching power off during the day can help save us power without losing functionality. This is accomplished by using a photoresistor to control the gate voltage in a common source amplifier. The output is then inverted and fed to two relays which control power to the entire device (less the daylight sensor).

2.2.4. Power Control Module: Debugging

The power control module worked immediately when we tested it in the lab. We had designed the sub-circuits using simulation results from LTSpice. We tweaked the value of R3 to make the daylight sensor work better and used standard resistor/capacitor values. Other than these minor modifications, the circuit did not change relative to its first revision.

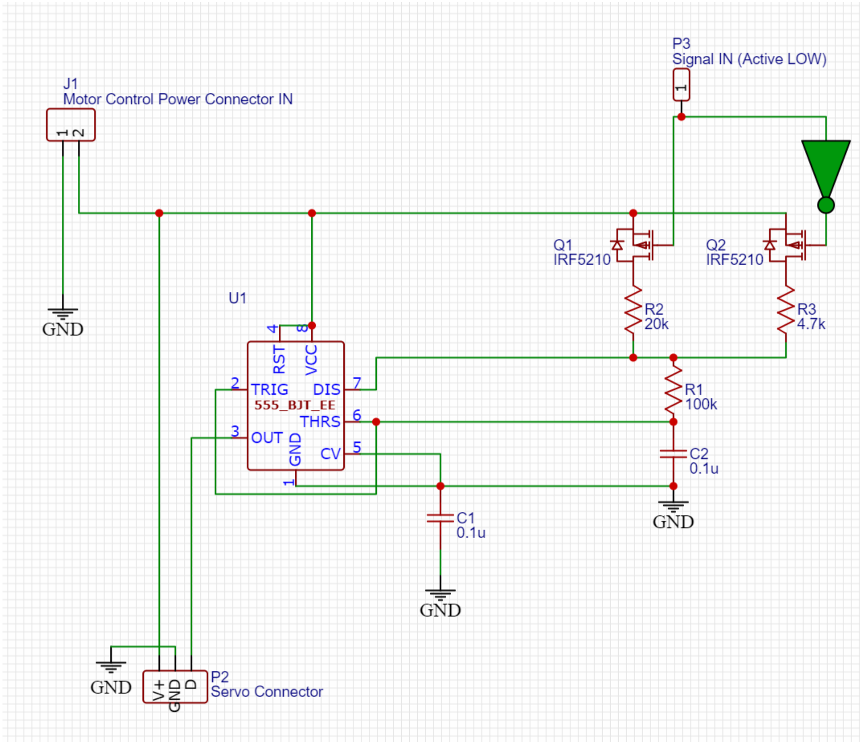
2.3. Motor Control Module

The motor control module is responsible for positioning the servo motor which physically disables the pressure plate in the trap. The servo must be placed at 0 degrees of rotation and 90 degrees of rotation. The servo maintains its position when powered off.

2.3.1. Motor Control Module: Purpose

The motor control module positions the servo to disable the pressure plate using an analog circuit. We chose to implement the motor control module in analog circuitry for two reasons. The first reason is to reserve computing power for image processing on the microprocessor. The second reason is because ECE 445 guidelines state that microprocessor use should be limited to computational intense tasks. Therefore, implementing this circuit in hardware was a good exercise in design.

2.3.2. Motor Control Module: Schematic



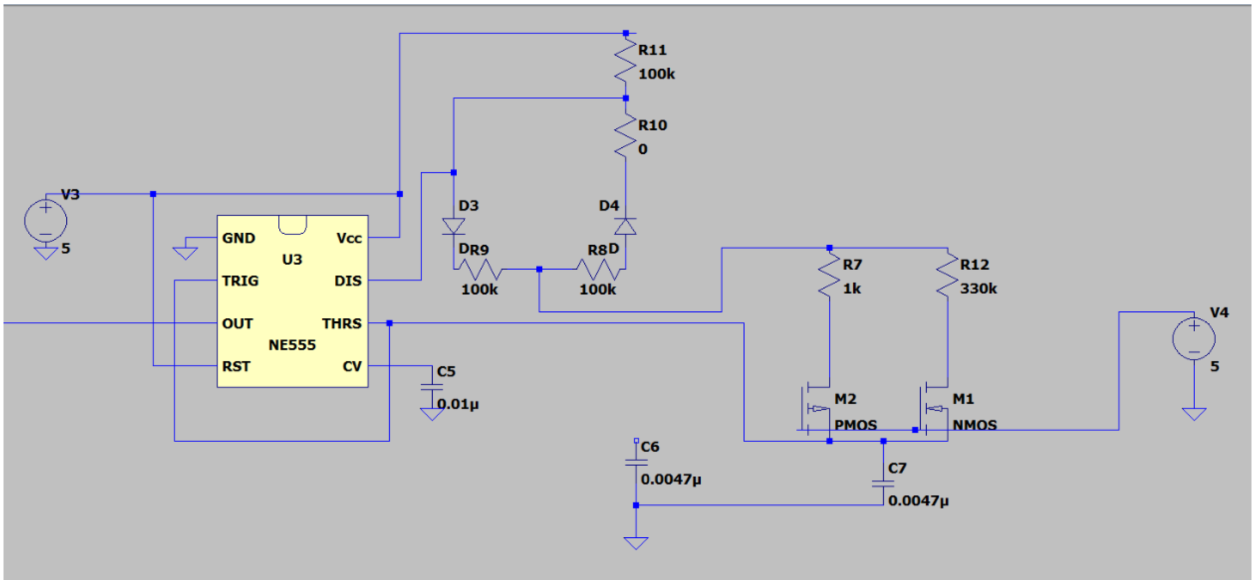
Above: Motor Control Module Circuit Schematic

2.3.3. Motor Control Module: Implementation

The final revision of the motor control module is pictured above. The circuit can output two different signals to the servo when powered on. The servo is positioned by varying the positive pulse width of a square wave. The 555 timer circuit creates this square wave, and using two PMOS devices allows us to choose between R2 and R3. In lab testing, this gave us a positive pulse width of approximately 900us and 1990us. This was enough to move the servo through the necessary range of motion. The inverter pictured in the schematic is the same one used in the power control module because these circuits were implemented on the same breadboard and PCB layout.

2.3.4. Motor Control Module: Design Considerations

Several designs were discarded when figuring out how to change the pulse width of the output waveform. The biggest hurdle we had was selecting the correct MOSFETs and determining a topology that allowed us to choose between the two resistors in the signal path. One of our earlier attempts is pictured below.



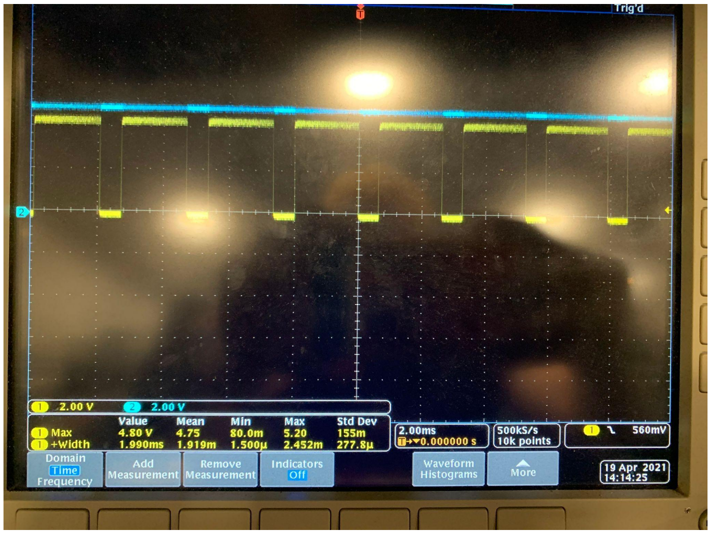
Above: Intermediate Revision of the Motor Control Module Circuit Schematic

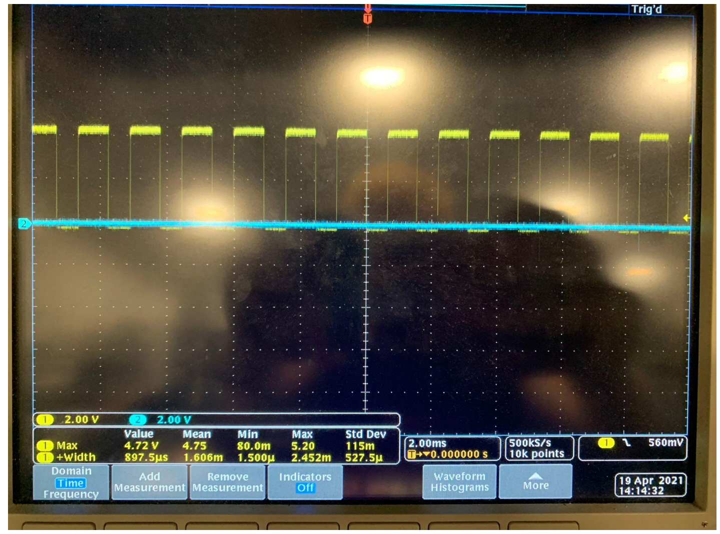
Although the topology is different overall, there is an issue with selecting between resistors R7 and R12 in this way. C7 pictured above charges and discharges periodically when the circuit operates. This affects the gate-source voltage of the transistors, allowing them to conduct at times when undesired. In our final working revision of the circuit, the PMOS gate-source voltage is referenced to the supply voltage. Therefore, we achieve switching of the PMOS devices that is not influenced by the signals propagating throughout the timer circuit

2.3.5. Motor Control Module: Debugging

Beyond designing a way to choose between pulse widths, we also had to figure out what pulse widths to send. Our original understanding of how the servo worked was incorrect. The data sheet is lacking information, therefore we expected the servo to respond to varying the duty cycle of a square wave at a particular frequency. In the lab, we found this to not be the case. This meant that our original circuit design did not work.

When debugging, we first controlled the servo directly with the function generator to find the pulse range of 800us to 2000us. We then simulated circuits in LTSpice to arrive at our final topology. We then breadboarded the circuit and realized that non-ideal behaviors such as transistor drain-source resistance affected the pulse widths that our circuit could output. We tweaked the resistor values on the breadboard to arrive at the final, working design. Two images showing the output waveforms can be found below. The blue line represents the switching signal. The positive pulse width is measured on the scope at the bottom.



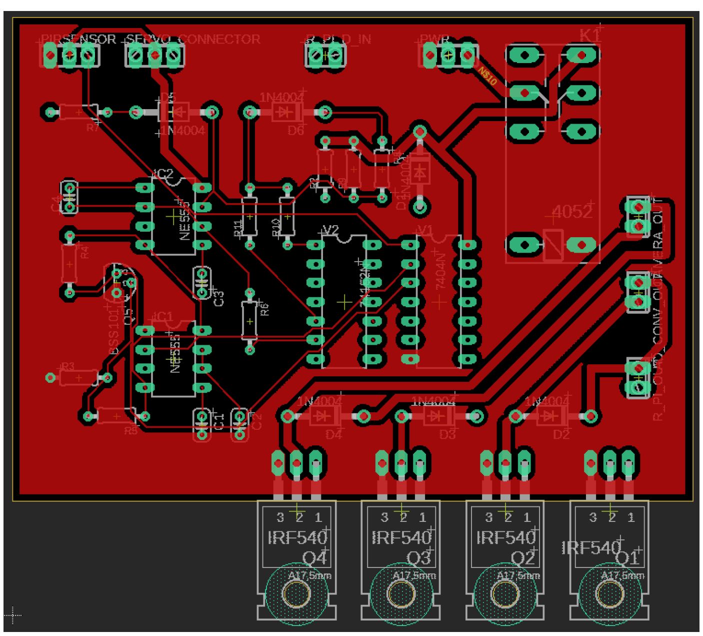


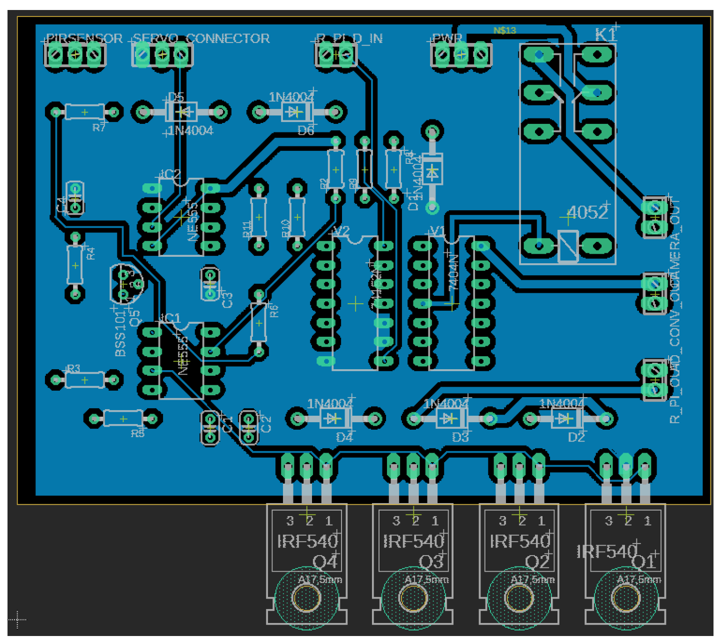
Pictured Left: Oscilloscope Reading of 1990us Pulse Width *Pictured Right: Oscilloscope Reading of 897.5us Pulse Width*

2.4 Printed Circuit Board

We designed a printed circuit board in April of 2021 for our project. The board contained the power and motor control circuits on one board. Although the board was manufactured, we received it too late in the semester to use it. Furthermore, the circuit board has the outdated topology of the motor control module printed on it. A redesign of the PCB is a future improvement for this project.

2.4.1. Printed Circuit Board: Topology





Pictured Left: Top Layer of PCB *Pictured Right: Bottom Layer of PCB*

2.4.2. Printed Circuit Board: Features

The most interesting design feature of our PCB is the use of two ground planes. The top plane is true ground for the power control module and would be connected to ground for the DC-DC converters. The bottom plane serves as ground for the motor control module sub-circuit. This ground plane is isolated from the true ground plane by an NMOS transistor. The transistor is switched when an animal is in proximity to the trap, as previously mentioned. Using two planes in this way made trace routing much easier. The power traces on the PCB were sized according to an online trace width calculator. [4] A safety factor was added to the power trace widths to arrive at 40mil.

Future improvements to the PCB layout should be made. The topology must be updated to reflect the latest working revisions of the circuits. Additionally, the signal traces should be increased in width from 6mil to 10mil. Finally, some signal traces should be re-routed. A few run very close to component pads and could be damaged during the soldering process.

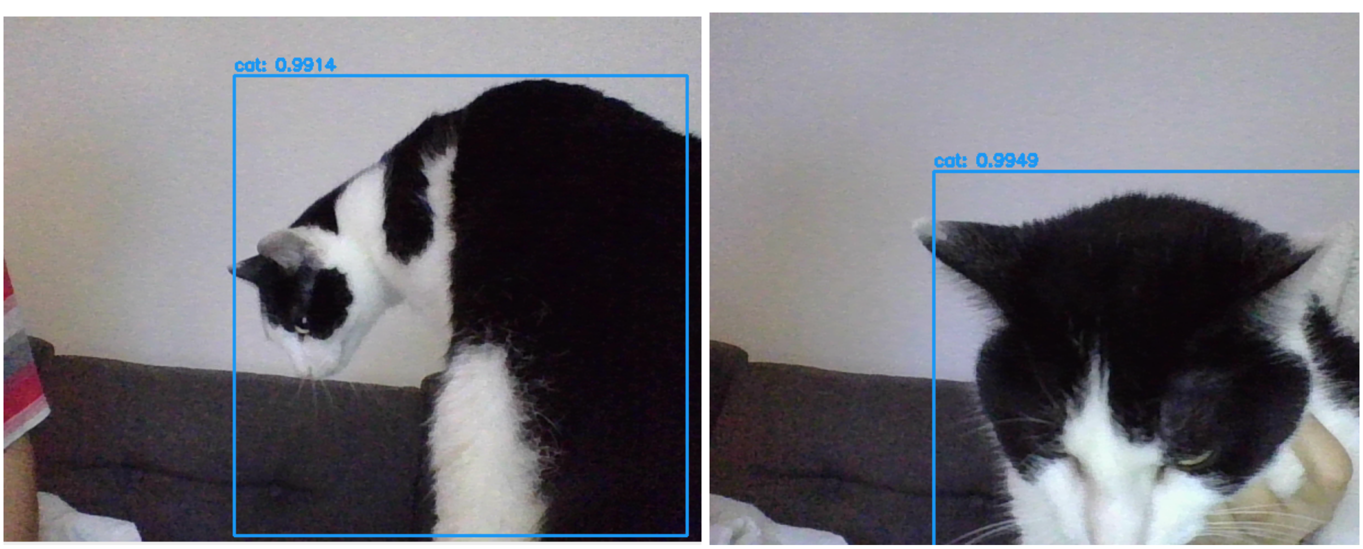
2.5. Computer Vision & Machine Learning

For the machine learning part, we built our program in the environment of Linux (Ubuntu 16.04). The language used is Python 3.7.4, with supporting libraries Tensorflow 1.14.0, Keras 2.2.4, Numpy 1.17.4. The application programming interface and database system is based on YOLO version 3. [5] The trained dataset and other related code are both original.

2.5.1. Computer Vision & Machine Learning: Purpose

The machine learning aspect of our project achieves two goals: animal detection and animal classification.

If any animal is detected through our night vision camera, a bounding box will square that animal with our prediction of its class. An example is shown below. The input signal comes from a real-time night vision camera, and the output image will be shown on the computer screen.



Above: Demo Screens with a cat through the real-time camera

2.5.2. Computer Vision & Machine Learning: Requirements

Our complemented machine learning program is fast, accurate, and convenient to use. Based on our recording, it takes 0.15-0.23 seconds for each frame of object detection. Detection speed is based on internet connecting speed. The picture below shows an example case of the average time of detecting each frame of videos. And based on a trained dataset, the accuracy of prediction can reach up to 70 percent. Input images are automatically captured from the in-time camera.

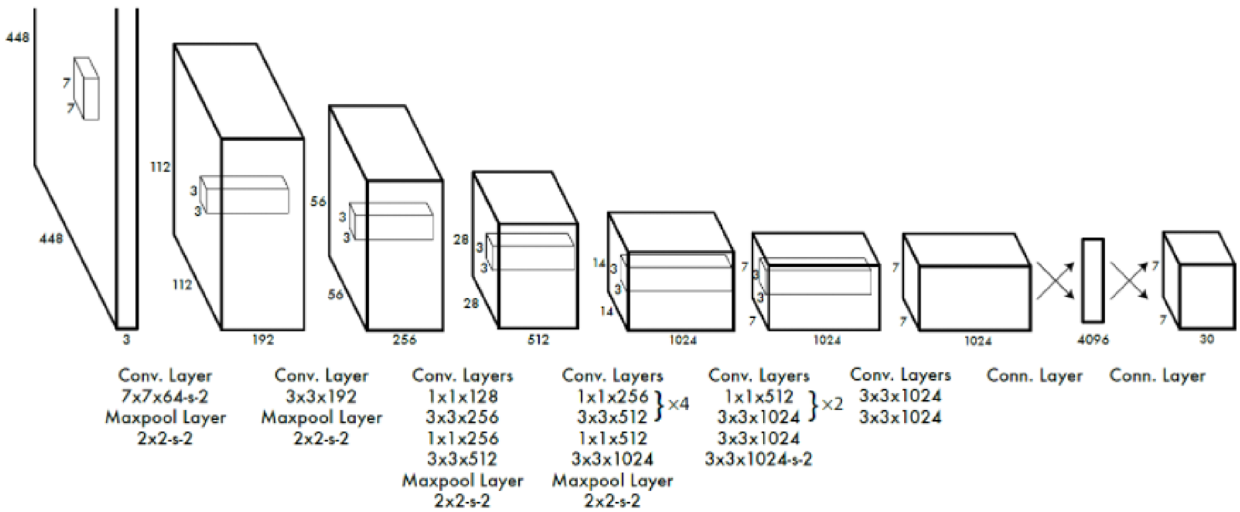
```
[INFO] Our SmatTrap ML took 0.167678 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.165931 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.164086 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167707 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.168516 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.168392 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167460 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167712 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.169045 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.170969 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.180642 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.175180 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.180270 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.176859 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167384 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.166099 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.163617 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.162779 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167140 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.161730 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.161266 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.169407 seconds to recognize this frame.
```

Above: An Example of Animal Detection Time

2.5.3. Computer Vision & Machine Learning: Processing

There are three steps needed to process the images captured by the camera.

The first step is to resize the input image into 448 x 448 grid cells. If animals are detected inside any grid cells, this grid cell will be responsible for this object. Then we run a single convolutional network on the desired image. Every grid cell predicts a number of bounding boxes, and each bounding box calculates its confidence number. The last step is to threshold the resulting detections by the module confidence.[6] We check for two values in the last step. The first value to check is the confidence value, which denotes how accurate our prediction could be. If a confidence value is less than 0.5, then the bounding box with the animal's predicted class will not be shown on screen. The second value to check is the NMS value(non max suppression)[7], which is used to filter out useless bounding boxes. If several bounding boxes have a similar overlapping area, NMS will filter them out and leave the most accurate bounding box shown on screen.



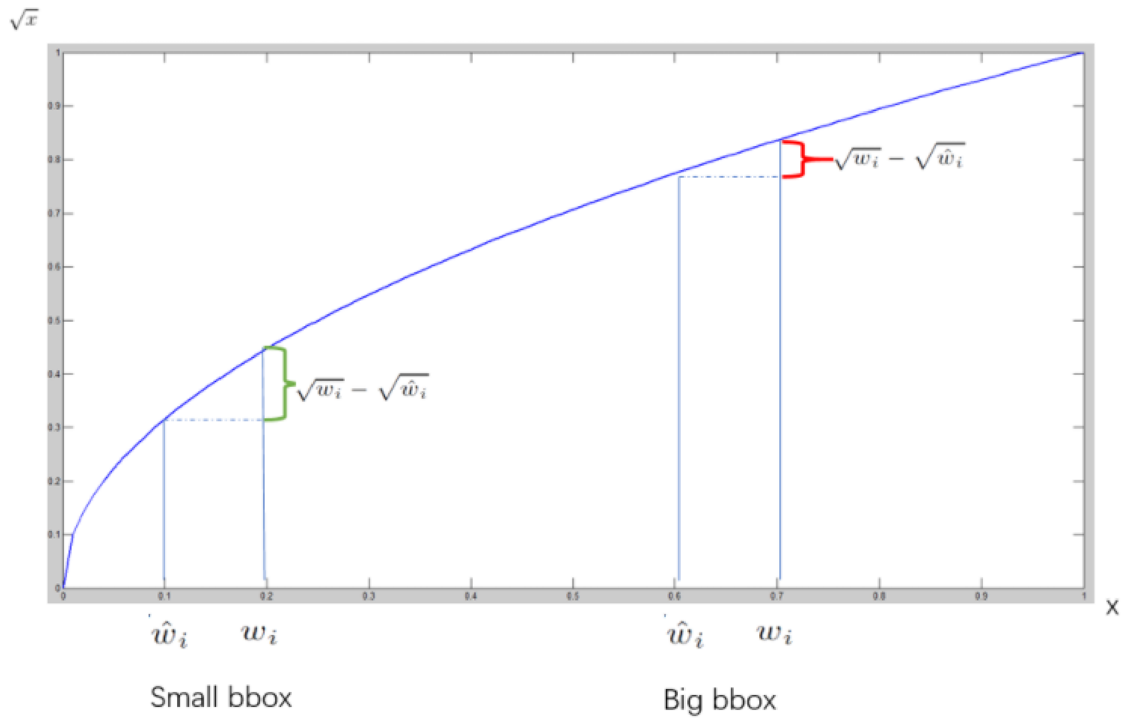
Above: Detailed Convolutional Network Process

2.5.4. Computer Vision & Machine Learning: Error Calculation

We find there is a close relation between size of bounding box and accuracy of class prediction.
In the image below:

X axis	=	*bounding box side length*
Y axis	=	*accuracy of our predictions*
Wi	=	*width of bounding box*
$\sqrt{x}$	=	*root of confidence value*

The relation between the width of a bounding box and the root of its confidence value is shown by the curve below. With the same difference in width of the bounding box, the difference of prediction accuracy is less affected in the larger bounding box (red brace), and is affected more in the smaller bounding box (green brace). Therefore, if the animal shown in the image is large enough, our prediction can be more precise. [8]



Above: Curve Showing the Relation Between Size of Bounding Box and Confidence Value

3. Design Verification

The state of our project at the end of the semester does not reflect the final design. Rather, the project is in a testing configuration. A switch block and an LED array allow us to simulate and view various signals throughout the hardware of the device. Therefore, we can test certain functions before all of the sub-systems of the project are integrated.

Beyond these internal debugging provisions, we also used lab equipment such as an oscilloscope to verify signals within the device. As pictured in section 2.3.5., the oscilloscope helped greatly when determining how to control the servo motor with an analog circuit.

At the time of our project's demonstration, design verifications were shown in several ways. LED indicators showed that the proximity-sensing feature of our device was working. In addition to switching an LED, the proximity sensing circuit also switched power to the motor control unit. The motor control unit was verified directly by actuating the servo. The servo was mounted to the cage and was physically tested to make sure that it could properly control the motion of the pressure plate. An image of the servo controlling the pressure plate can be seen in section 5.1.. The daylight sensing circuit was verified only with an LED indicator. The daylight sensing circuit did function as designed, but the output was not hooked up to the relays as seen in the schematic in section 2.2.2..

The computer vision was verified in the lab by using image prints held in front of the camera. This verification has its limitations, but served as a good checkpoint. The images allowed us to learn how profile and direct views of animals can be detected. After the lab demonstration day, the software was tested using a free-roaming (non-caged) pet cat. As seen in section 2.5.1., the software was able to confidently detect the cat. Unfortunately, we were unable to test the software on outdoor animals at this stage. In further development of this product, this testing would be essential.

Our initial requirements and verifications table can be seen in the appendix. There are two notable shortcomings of our project. The first is that the system does not power off when daylight is detected. This is simply solved by incorporating the relays in the design, which we already have in inventory. The second shortcoming is the lack of integration between the computer vision and hardware components of the project. This integration is discussed in section 5.4., *Future Work*.

4. Cost

This section details the costs associated with this project. The total cost of development for the project including parts and labor is $9,040.

4.1. Parts

A table is shown below describing the parts used, their quantities, and their costs. Costs are rounded to the nearest dollar amount unless the cost is less than one dollar. The total parts cost is $220. This cost includes some spare parts for making a second unit or for troubleshooting the existing design.

Part Name	Brand & Part Number	Quantity	Cost (Total)
Raspberry Pi 3 B+	1690-1025-ND	1	$45
Weatherproof Color Security Camera With Night Vision	Harbor Freight 95914	1	$30
PIR Sensor	Adafruit Product #189	1	$10
Photoresistor	Adafruit Product #161	1	$1
Analog to Digital Converter (RCA to USB for camera)	BR116-US	1	$12
90 degree Servo Motor	Adafruit Product #1143	1	$10
Ryobi 18V Battery	Ryobi P108 or similar	1	$42
18V to 5V DC converter	5A6A7A8A10A	1	$5
555 Timer	Sparkfun COM-16473	2	$2
P-Channel MOSFET	F953ON	2	$20 for 10 Pcs.
N-channel MOSFET	IRFZ44	5	$8 for 10 Pcs.
Resistor Kit	Sparkfun COM-10969	1	$8

0.1uF Capacitor	Sparkfun COM-08375	3	$0.75
4.7uF Capacitor	100YXF4.7MEFC5X11	1	$0.31
Hex Inverter	Texas Instruments CD74AC04E	1	$1
Pressure Plate Activated Animal Cage	Rural King 75500014	1	$25

4.2. Labor

Labor costs are estimated using the formula:

(Hourly Salary) x (Actual Hours Spent per Team Member) x 2.5 x (Number of Team Members)

Using an hourly salary of $28, three team members, and 42 hours spent working per member, the above formula yields a total labor cost of $8,820.

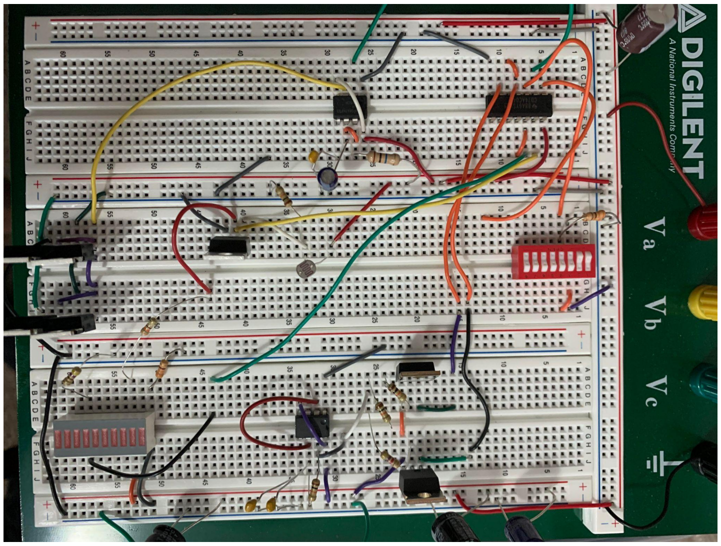
5. Conclusion

As mentioned previously, the current state of the project does not reflect the final design revisions. This section details the accomplishments of the project so far, the known issues, and how future work on the project should proceed.

5.1. Accomplishments

In the default state, the servo arm is able to prevent the pressure plate from triggering the trap regardless of the creature on it. When a signal representing a targeted creature is sent, the servo arm rotates allowing the creature to trigger the trap. The PIR sensor is able to detect motion within a cone in front of the trap, and send a signal indicating that motion has been detected for a minute. The object detection software was able to distinguish between a targeted creature (raccoon) and a non-targeted creature (cat) from different angles during the daytime.

Pictured below is the current state of the hardware and the trap addon. Note how the servo arm touches the pressure plate, preventing it from being depressed.



Pictured Left: Current State of the Circuit on Breadboard

Pictured Right: Servo Restricting the Trap's Pressure Plate

5.2. Known Issues

Our project has two known issues that need to be solved before advancing. The first problem is that the servo motor sometimes has jitter, meaning that it twitches about 2-4 degrees of rotation. We believe this to be a power supply issue and have attempted to solve it with capacitors across the power supply. This approach has not fixed the issue. Next debugging steps include monitoring the servo signal waveform to see if it is not providing a constant pulse width.

The other known issue with our project is the nighttime accuracy of our computer vision. This is solved simply by collecting more training data at night and training the machine learning algorithm with that data. Low-light performance is not entirely missing in the current state of the project, but confidence levels are greatly diminished in low-light settings. This is mainly because color information is lost at night.

5.3. Ethical Considerations

No animals were caged or trapped in the development and testing of our project. The only animal used was a pet cat. The cat was free-roaming the owner's home when images were captured as seen in section 2.5.1.. Any future testing process must follow the guidelines set forth by the USDA Animal Welfare Act and the NIH Public Health Service Policy as stated by Illinois Institutional Animal Care and Use Committee.

No hazardous or volatile materials are used in our project. All components we will use are consumer grade and can be purchased legally online. Furthermore, all group members have completed lab safety training according to the safety guidelines of The University of Illinois.

5.4. Future Work

The main thing that needs to be accomplished for our project is to localize the computer vision system, and improve the creature detection to work during the nighttime. This would be done by replacing the laptop with a microprocessor and localizing the creature detection software. Additional parameters would have to be added to the creature detection software to detect with infrared light.

Another thing that needs to be done is to implement daylight switching to our project. This would be done by interfacing the photocell resistor sub-circuit with relays to shut power to the rest of the rest of the circuit whenever it is daytime.

The circuit components still need to be interfaced with the microprocessor in order for the project to function properly. The computer vision system has to be interfaced with the rest of the circuit by hooking it up to the motor control unit to send a control signal when a targeted creature is detected. The power supply for the project which consists of the 18V battery and the voltage converters still has to be connected to supply power locally.

Lastly, the circuit components have to be transferred from the breadboard to a revised PCB.

6. References

[1] “The Disadvantages Of Using Live Traps”, Wildlife Whisperer, Available: https://totalwildlifecontrol.com/about-us/disadvantages-of-trapping/. [Accessed 1 April 2021].

[2] ElectronicsTutorials, "555 Timer Tutorial," [Online]. Available: https://www.electronics-tutorials.ws/. [Accessed 7 April 2021].

[3] ElectronicsTutorials, "MOSFET as a Switch," [Online]. Available: https://www.electronics-tutorials.ws/transistor/tran_7.html. [Accessed 7 April 2021].

[4] Advanced Circuits, "PCB Trace Width Calculator," [Online]. Available: https://www.4pcb.com/trace-width-calculator.html. [Accessed 7 April 2021].

[5] YOLOv3: An Incremental Improvement, Redmon, Joseph and Farhadi, Ali, [Source code]. Available: https://pjreddie.com/darknet/yolo. [Accessed 2018].

[6] “How Do Convolutional Layers Work in Deep Learning Neural Networks?”, Jason Brownlee, Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks. [Accessed April 17, 2019]

[7] “Selecting the Right Bounding Box Using Non-Max Suppression”, Aishwarya Singha, Available: https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation. [Accessed August 3 2020]

[8] “A Gentle Introduction to Object Recognition With Deep Learning”, Jason Brownlee, Available: https://machinelearningmastery.com/object-recognition-with-deep-learning/. [Accessed May 22 2019]

7. Appendix

7.1. Requirements & Verifications Table

Requirements	Verifications
Trap should have a minimum of twelve hours of battery life and send power to all devices.	1. Make sure the device has been optimized not to waste too much power on components like resistors by testing the circuit with different values. 2. When it is nighttime, minimize power consumption by shutting down the camera unless the PIR sensor detects a creature. 3. Minimize power by having the camera shut down after 30 seconds if the detected creature was not a targeted creature.
The trap should make a decision within ten seconds of the animal entering the cage.	1. Optimize the time it takes for the camera to turn on and start sending the image data when a creature is detected via the PIR sensor. 2. Optimize the machine learning algorithm to perform object detection in about 1-2 seconds per frame.
The Motor Control Unit must be able to rotate the servo through its full range of motion	1. Verify ideal resistor values using LTSpice 2. Ensure that the pulse waveforms fall within the specifications of the servo motor 3. Test servo motor with a function generator to see which duty cycle and frequency will create the desired rotation 4. Compare the simulated waveforms and lab results and modify the simulated circuit as needed 5. Build the circuit on a breadboard and test with an oscilloscope to ensure that the proper outputs are achieved in hardware 6. Test the physical circuit with the servo to ensure full movement
The Power Control Module must be capable of switching power on to other components for one minute after receiving a pulse from the PIR sensor.	1. Simulate the circuit design in LTSpice and verify that the preliminary design can create a boolean true signal with a duration of one minute when an input pulse is received. 2. Record the resistor and capacitor values necessary for this behavior and source them for building the physical circuit. 3. Obtain the PIR sensor and measure the real output waveform. 4. Return to simulation and emulate the output from the PIR sensor. Ensure that the circuit still functions. 5. Build the circuit on a breadboard. Use the function generator as an input and measure the output.

	6. Use the PIR sensor as an input to the circuit. Control the sensing environment with a box and excite the sensor. Ensure that the measured output pulse lasts for one minute.
The entire system should turn off during the day.	1. Simulate the light detecting circuit using LTSpice. Ensure that the sensing circuit can output a different boolean value when the light sensor is at its minimum and maximum resistance values. 2. Obtain the photoresistor and measure its actual resistance values during a typical night and typical day. 3. Return to simulation and adjust the circuit now knowing the real world values. Include a proper tolerance with reference to the measure values obtained in step two. 4. Build the circuit on a breadboard. Test the output using a resistor to emulate the photoresistor. 5. Now use the photoresistor. Use a battery to power the circuit, connect the output to an LED, and test it in outdoor environments. Ensure proper functionality. 6. Test the circuit to make sure that it can adequately power the relays which switch the main power to the device. If this approach consumes ample power, consider using MOSFETs instead.