

Department of Electrical Engineering and Computing Systems (DEECS)

Interactive Penguin Fish Dispenser

Team #CE16

**Kristen Diaz
&
Nathan Patterson**

April 26, 2019

Submitted in partial fulfillment of the degree of

Bachelor of Science

Computer Engineering

Team member's Signature (Kristen Diaz) *Kristen Diaz*

Team member's Signature (Nathan Patterson) *Nat Path*

Technical Advisor's Signature (Dr. Dieter Vanderelst)

Vanderelst

DEDICATION

Thank you to -

Dr. Dieter Vanderelst, our technical advisor for being supportive through our ups & downs, and letting us borrow equipment and chase you down.

Dr. John Layne, for giving us space in the Biology Department where we could test dead fish dispensing, without anyone getting made at us about the smell. Also thank you for letting us borrow equipment when needed and being there for advice.

Stephen Mergner, for being our mechanical superhero/godfather, modifying and cutting parts, and giving us endless advice on how to build things.

Dr. Katie Kalafut & Rickey Kinley, for sponsoring our project with the Cincinnati Zoo and believing us through the thicks and thins of the project.

TABLE OF CONTENTS

Abstract	5
INTRODUCTION	5
Problem	6
Rationale/Benefits	7
Credibility	8
Project Goals	8
DISCUSSION	9
Project Concept	9
Design Objectives	11
Methodology/Technical Approach	14
Electrical Development	14
Mechanical Development	18
Budget	22
Timeline	23
Problems Encountered/Analysis of Problems Solved	25
Future Recommendations	25
CONCLUSION	27
References	28
Appendices	29
Appendix A: Arduino Code	30
Appendix B: Component Manuals	33

LIST OF FIGURES

Figure 1: An Example of a Penguin with Bumblefoot	6
Figure 2: Graphs of 5 Penguins' Progress with Enrichment	7
Figure 3: Objective Tree	10
Figure 4: Table - Design Specification Table	11
Figure 5: Table - Metric Chart	11
Figure 6: Outline of Mechanical Set-up	13
Figure 7: Black & Glass Box Diagrams of Design	14
Figure 8: Ozonium Waterproof 12V Power Supply	15
Figure 9: Arduino UNO	15
Figure 10: L298N Motor Driver	15
Figure 11: 12V DC Motor	16

Figure 12: Toggle Switch	16
Figure 13: Electrical Circuit of the System	17
Figure 14: Picture of Components in the Electrical Box	17
Figure 15: Fish Dispenser Flap Design	18
Figure 16: Fish Holding Hopper	19
Figure 17: 150-300 GPH Submersible Pump	19
Figure 18: Metal Base Measurements	20
Figure 19: PVC System for the Toggle Switch	21
Figure 20: Table - Build of Materials	22
Figure 21: Table - Estimated Labor Costs	23
Figure 22: Gantt Chart Timeline	24
Figure 23: Picture of Two of the Little-Blue-Foot Penguins at the Cincinnati Zoo	27

ABSTRACT

Dr. Katie Kalafut at the Cincinnati Zoo has a hypothesis that the small blue-foot penguins are having health problems such as bumblefoot (pododermatitis), a bacterial infection of the feet, due to not swimming as much in captivity as they would in the wild to hunt for food. Previous projects to create systems to lure the penguins into the water to swim more often have been not seen much success in the zoo's experience. An interactive fish dispenser would help resolve this problem by forcing the penguins to swim into the water in order to interact with the system, and then dispense a fish as a reward.

INTRODUCTION

The purpose of the report is to describe the current problem of penguins at the Cincinnati Zoo needing additional interactive enrichment to help improve health issues with the penguins and the solution, implementation and benefits. The Cincinnati Zoo has a need for additional interactive enrichment to increase the quality of life for their penguins. This includes enticing the penguins into their pool for environmental interaction. Past attempts and projects to increase the penguins' interactive enrichment have been not as successful as needed for the zoo. The scope of the project is to create an interactive enrichment tool that will successfully get the penguins in the exhibit to go into the water, swim, and retrieve a fish as a reward.

Problem

At the Cincinnati Zoo the zookeepers noticed that the penguins are not swimming as much in the water as they otherwise normally would in the wild. This can negatively affect a penguin's overall physical and mental health. Studies have shown that infections such as bumblefoot (pododermatitis), a bacterial infection of the feet, can possibly be decreased with increased environmental enrichment such as swimming[2]. Not only would penguins benefit from an increase in interactive enrichment, but also zoo keepers will have a tool to be able to encourage penguins to swim in the water and guests will be able to watch the swimming penguins. An effective interactive enrichment tool for penguins at the Cincinnati Zoo could then also be implemented at other zoos and penguin rehabilitation centers to increase the quality of life for penguins around the world.



Figure 1: An example of a penguin with bumblefoot [7]

Rationale/Benefits



Figure 2: Graphs of 5 Penguins' progress with Enrichment

The graphs above show the horizontal and vertical lesion sizes in both infected feet over the course of twelve weeks through the enrichment process on five penguins. The enrichment process consisted of the penguins being stimulated to remain in the water one more hour daily with aquatic enrichment. The graphs show that over the course of the enrichment the lesion sizes in four out of the five penguins were drastically reduced. Penguin five's lesions began to reduce after the twelfth week. [6].

The benefits of this penguin enrichment project are that it could reduce the effects and causes of bumblefoot that penguins in captivity experience. Getting the penguins to spend more time in the water lessens the time they spend on the hard, rough surfaces, which in turn can reduce any discomfort in their feet. Another benefit of this enrichment is that making the penguins go into the water to get their own food simulates what it would be like for them if they were still in the wild. Aside from the penguins' benefit, the Cincinnati Zoo and the zoo visitors will benefit from this as well. The Cincinnati Zoo will not have to worry about their penguins not swimming as much and possibly getting infections. The zoo visitors will benefit by being able to experience how penguins act and retrieve food in the wild.

Credibility

Kristen Diaz & Nathan Patterson are both Computer Engineering students at the University of Cincinnati with previous work experience in wiring and design. Kristen Diaz has experience in designing and building interactive demos for industrial grade manufacturing sensors and equipment. Nathan Patterson has experience in installing and wiring interactive equipment for schools and businesses.

Project Goals

The project proposal is to construct, wire, and program an interactive enrichment tool for the Cincinnati Zoo's penguin exhibit that will increase penguin activity in their pool to aid in the increase of the penguins' physical and mental health. This includes having the tool/project be penguin friendly as to not put the penguins in any possible harm, which might include them getting stuck or hurt by a part of the tool while it is in the exhibit. The project will also be guided by Dr. Katie Kalafut and Rickey Kinley who work with the penguins and specialize in penguin behavior at the Cincinnati Zoo in addition to the technical advisor.

DISCUSSION

Project Concept

The main concept is to create a system that the penguins can safely interact with that dispenses a fish. This includes having the system be durable enough for everyday use to withstand and water or weather that it may be exposed to. The below project goals outline these reliability and safety goals in order for a successful overall project. These are also diagrammed in the Objective Tree in Figure 3. The utmost goal being that the system is safe for the penguin, and at no time poses a safety risk.

Project Goals

- Reliable
 - Should get the penguins to swim more
 - Dispense fish for penguins
 - Durable
 - Waterproof
 - Weatherproof
 - Should work in indoor and outdoor enclosures
 - Moved between enclosures
 - Can be mounted/stabilized in enclosures
- Safe (for the penguins)
 - Penguins cannot get caught
 - Penguins cannot be able to chew on the design
 - Penguins should not be hurt by fish
 - Fish should be reachable for the penguins
 - Fish must be kept fresh so no penguins get sick

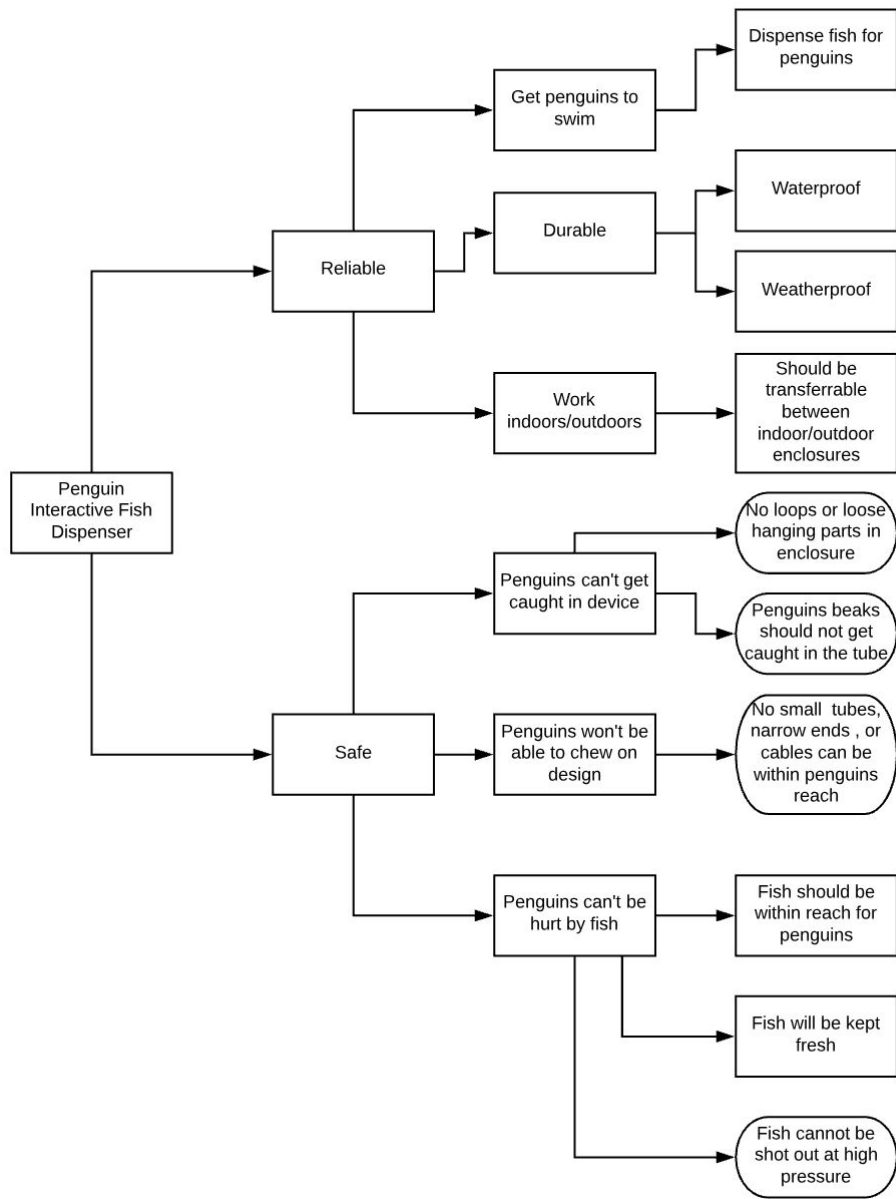


Figure 3: Objective Tree

Design Objectives

The following Design Specification Table (see Figure 4) and Metric Chart (see Figure 5) help outline the main functions and design objectives that the overall system needs to follow in order to follow the concept design.

Functions	Performance Specifications
Display fish as an output	The design must allow the fish to be displayed in a safe manner for the penguin to retrieve it.
Toggle switch used to detect penguin interaction	The design must detect when the penguin interacts with the device.
Fish are kept fresh inside a container for a few hours	The design must allow fish to be stored and kept cool while inside the device.
Existing RFID set-up will be used to monitor penguin activity.	The design must allow for RFID detection to identify the penguins using the device.

Figure 4: Table - Design Specification Table

Objectives	Metrics
1. Should get the Penguins to Swim More	Number of penguin interactions recorded by the RFID reader
2. Durable	Conditions where the system continues working safely under water or weather conditions
3. Work in the outdoor & indoor enclosures	Effective mounting locations in both the indoor and outdoor locations the system can be safely mounted
4. Penguins cannot get caught	Number of possible locations a penguin could get caught
5. Penguins cannot chew on parts	Number of edges/ends that could be possible chewed on
6. Penguins cannot be hurt by the fish	Average speed of the fish upon delivery & location

Figure 5: Table - Metric Chart

Once the design objectives were considered, the additional following constraints, functions, and means were put into the design to ensure the design objectives were met.

Constraints:

- Fish should not be shot out at a high pressurized force
- No loops or loose hanging part in the enclosure
- No small tubes or narrow ends should be in the enclosure within penguins reach
- All cables must be securely restrained out of penguin reach
- Beaks of the penguins should not get caught in the fish tube

Functions:

- Displays fish as an output for the penguins
- Must have an Interactive Input to detect penguin interaction
- Fish are held in a container where they stay at a fresh temperature for at least a few hours
- Existing zoo RFID set-up will be paired along with the system to record penguin activity

Means:

- Use Oregon RFID systems (what the zoo currently uses) to implement RFID with the system
- Use a tube system (hole-in-the-wall) to hold the fish output
- Use a cone system to load and hold the fish (before dispensed)
- Use Arduino UNO to program system to dispense fish
- Use toggle switch to detect penguin interaction
- Use ice compartment to keep fish fresh

The previous mentioned design concepts were combined together to create the below diagram (Figure 6) of the mechanical set-up for the system. The diagram shows the fish holding area (hopper) that would be used to hold and store fish with ice. Then the arduino and the rest of the electrical components would be securely contained in an area connected to the system. A tube would come out from the bottom of the hopper, where a flap would be controlled by a DC motor to control the flow of fish. Another smaller pipe system would come down from the electrical component box, containing wiring for the toggle switch, which will go underneath the water for the penguins to be able to swim and hit with their body, activating the system. The zoo's RFID system is also shown, where after the dispensing system is added, a RFID reader could be installed near the toggle switch to detect penguin interaction, as all of the penguins have an RFID chip clipped onto their wing.

Outline of Mechanical Set-up

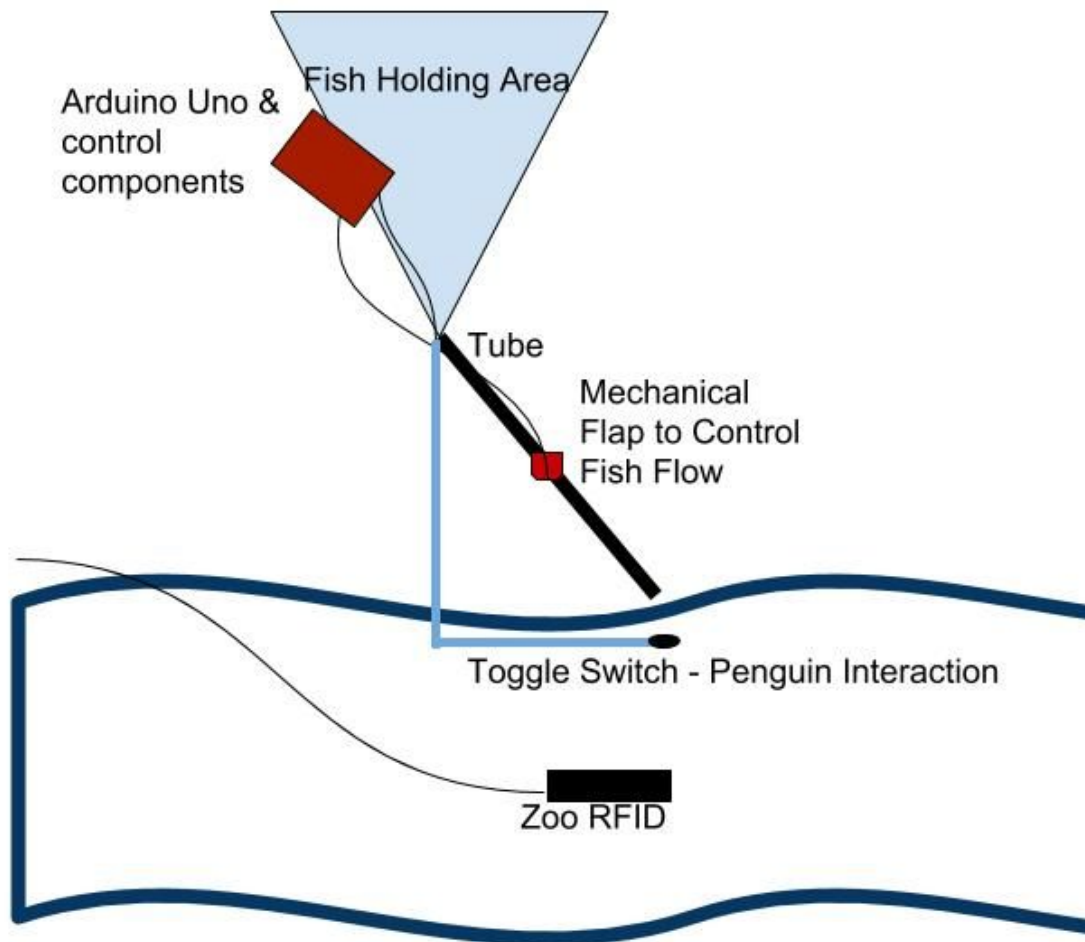


Figure 6: Outline of Mechanical Set-up

In Figure 7, the Black & Glass Diagrams are shown to give an additional overhead view of the different components and how they will work together. Showing the input signal from the toggle switch, the power for the electrical & mechanical components, and the fish from the holding area coming together, which will only show a single fish dispensed as an output.

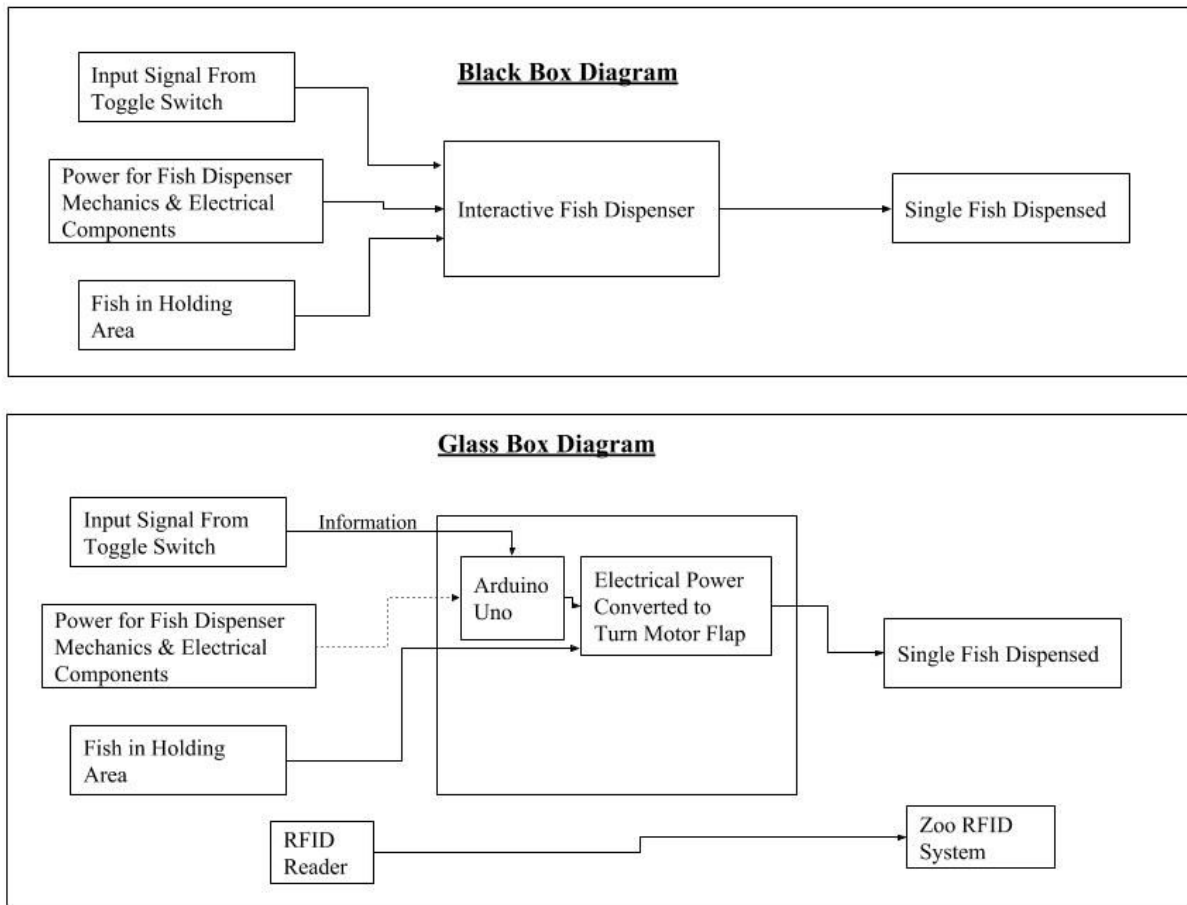


Figure 7: Black & Glass Box Diagrams of Design

Methodology/Technical Approach

During the construction and design stages, the team searched for similar dispensing systems that may be on the market commercially, however none were found. Thus the project was tested in phases focusing on the electrical and safety objectives first for achievability.

Electrical Development

To have a reliable power source, a waterproof transformer to create a reliable and safe 12VDC power supply for the rest of the components. The goal of only having one power supply, was so that during use, zoo keepers would only need to plug-in one cable to power the entire system.

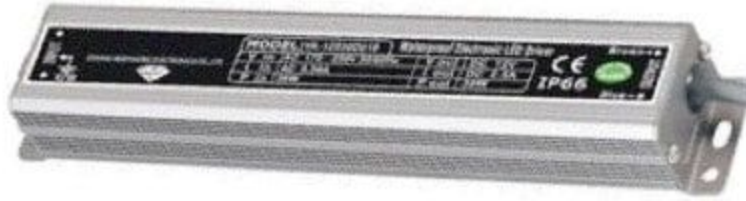


Figure 8: Ozium Waterproof 12V power supply

The “brains” of the system was an Arduino Uno that was used with an L298N motor driver to control a DC motor to control the fish dispensing flap.



Figure 9: Arduino UNO

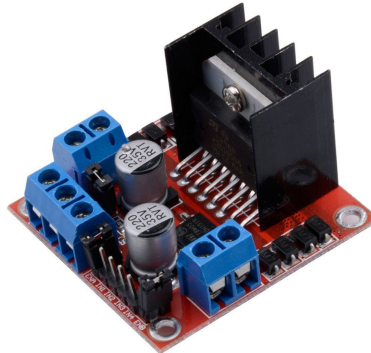


Figure 10: L298N Motor Driver



Figure 11: 12V DC Motor

The toggle switch used (shown in Figure 12) has a minimum requirement of 12V, and is commercially sold as a replacement toggle switch for jet skis.



Figure 12: Toggle Switch

The electrical circuit of the system can be seen in Figure 13 below. The components' connections are soldered in the system with a PCB board. A 330k Ω and 220k Ω resistors are used to create a voltage divider to step down the toggle switch voltage from 12V to around 4.8V. This was due to the toggle switch needing at least 12V for an accurate signal to be produced and read, and that the Arduino UNO input has a maximum voltage of 5.5V. A .5mA fuse is also inserted in-line with the source voltage of the DC motor, to protect the system from overdrawing current if something malfunctions with the motor.

A 9V voltage regulator was also added into the system to power the Arduino UNO with 9V instead of the straight 12V supply. This was due to concerns of additional heat being produced from the Arduino's voltage regulator if powered at 12V, because the Arduino's maximum power input voltage is 12V. The electrical components are all contained within an electrical box that is tight sealed, thus overheating could occur if extra heat is produced within the box.

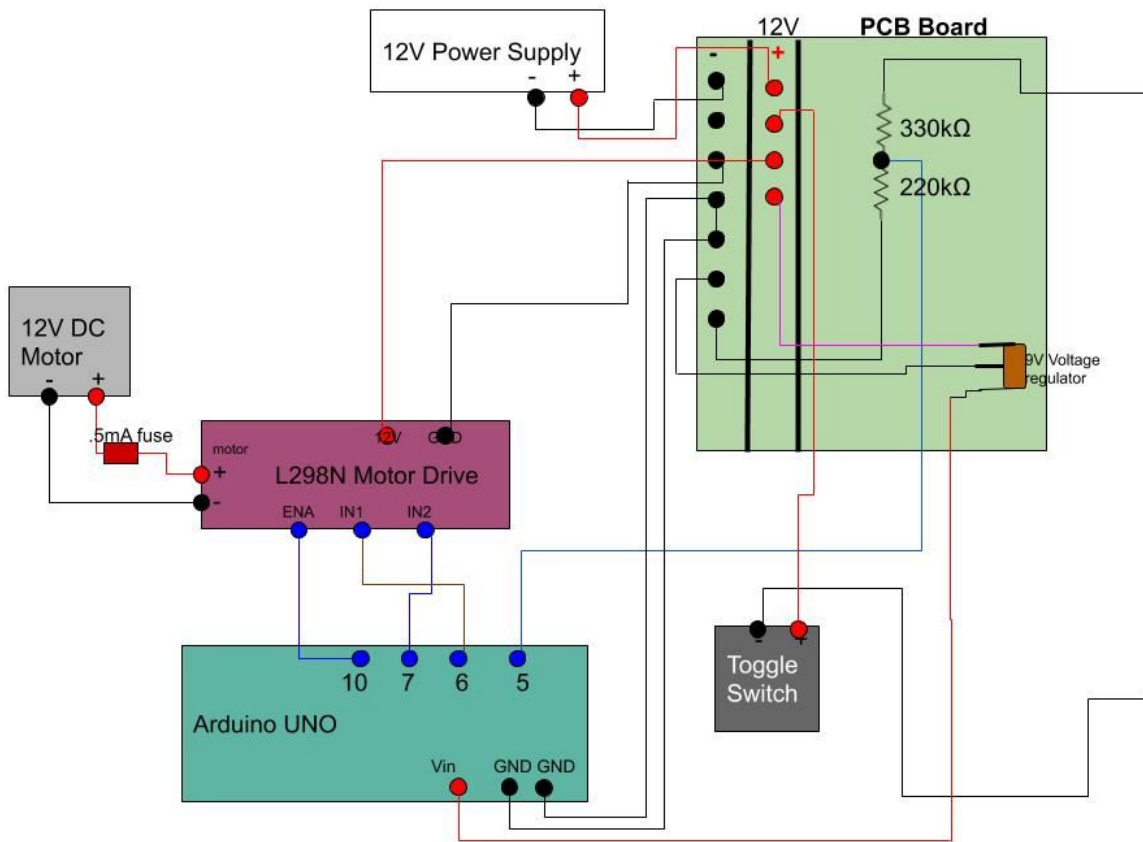


Figure 13: Electrical Circuit of the System

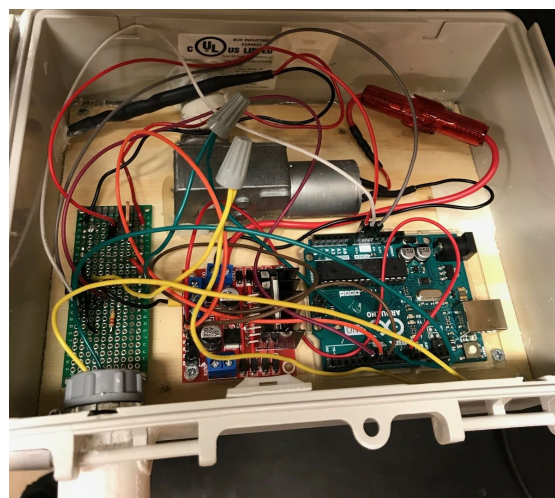


Figure 14: Picture of Components in the Electrical Box

Mechanical Development

For the mechanical design of the system, the fish dispenser flap was produced via a formlabs 3D printer. The CAD and STL models were created using Siemens NX12. Multiple variations of different flap designs were tried, and the final chosen flap design can be seen in Figure 15. A very small amount of JB Wled - WaterWeld was also applied on the back of the flap design after testing to help stop the flap before it hit inside the tube as it turned.

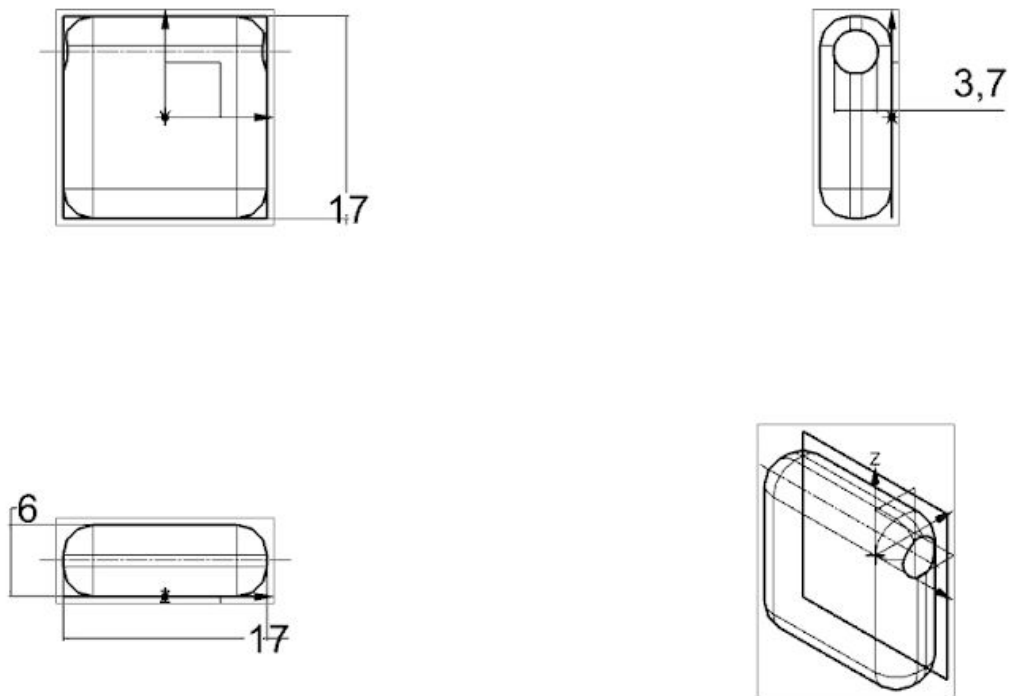


Figure 15: Fish Dispenser Flap Design

The fish holding area or hopper was a drain tank as seen in Figure 16, in order to have a professional looking somewhat-cone-shaped container.



Figure 16: Fish Holding Hopper

The main structure of the system was made out of steel slotted angle beams that were cut to 14" long and wide to support the hopper, then 30.5" tall, the structure and measurements can be seen in Figure 18. To create a sharper cone shape for fish flow- clear plastic lining is shaped into a cone and placed into the hopper. This space between the cone lining and the inside of the hopper is then used to store ice to keep the fish sitting in the hopper at a cool temperature.

For additional assistance with help fish flow through the system, a 150-300 GPH Smartpond fountain submersible pump seen in Figure 17 was added. The submersible pump can pump water from the penguin exhibit up into the hopper to help guide fish down the tube system.



Figure 17: 150-300 GPH Submersible Pump

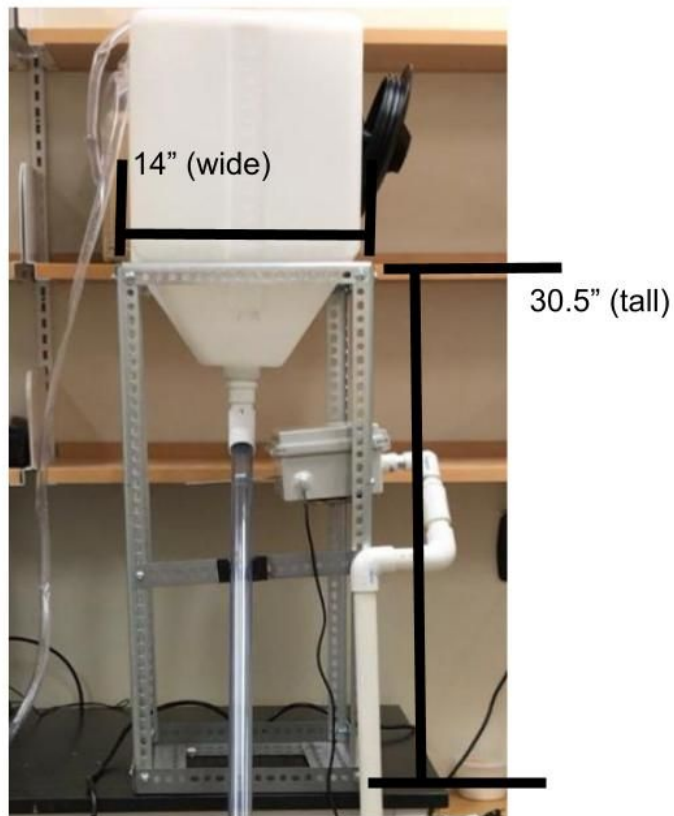


Figure 18: Metal Base Measurements

Coming out of the hopper, is a 1" PVC threaded coupler which transitions to a 1" PVC elbow, and then the clear 1" PVC tube. This clear PVC tube, is the where fish will travel out of before going into the water for a penguin. There is also a 1/8" steel rod that is attached to the DC Motor, where the flap is attached and turned from, that goes through this clear PVC tube.

Figure 18 also shows the electrical control box that is bolted on to a steel shelf to the metal base. Any openings or holes created of the box are all watertight sealed with waterproof sealants such as Dynaflex230, and JB Weld WaterWeld, to prevent any moisture from leaking into the electrical box.

In Figure 19, is the PVC pipe system for the wiring for the toggle switch. The PVC starts as a ½” hole from inside the electrical control box and the PVC components are in the following order:

1. ½” PVC threaded coupler
2. ½” PVC pipe, 1” long
3. ½” to ¾” PVC 90° elbow
4. ¾” PVC pipe, 1” long
5. ¾” to 1” PVC converter
6. 1” coupler
7. 1” PVC pipe, 1” long
8. 1” PVC 45° elbow
9. 1” PVC pipe, 3” long
10. 1” PVC 90° elbow
11. 1” PVC, 3’ long
12. 1” PVC 90° elbow
13. 1” PVC pipe, 2” long
14. 1” to 2” PVC converter
15. 2” PVC converter
16. 2” PVC end cap

Then the Toggle Switch sits an opening made in the 2” PVC end cap, secured by nuts. Then a ¾” PVC pipe was sealed at one end with JB Weld Water Weld, and then permanently fixed to the original toggle using JB Water Weld. The gaps between the now extended toggle and the original toggle were sealed with aquarium sealant.

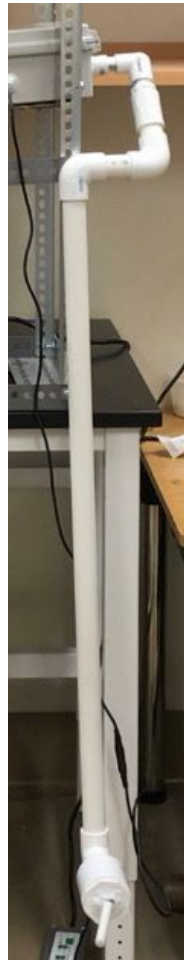


Figure 19: PVC system for the Toggle Switch

Budget

Figure 18 shows the build of Materials of what was bought for the project, electrical items are highlighted green for ease of reading. Figure 20 shows approximated labor costs for the project if at the rate of \$30/hour for planning steps and then for the construction & testing period. See the Gantt Chart for a further breakdown of project plan.

Build of Materials			
Item	Quantity	Item Price	Total
Waterproof LED Transformer - Outdoor AC Adapter	1	\$39.99	\$39.99
Clear Standard-Wall Unthreaded PVC Pipe - 4ft.	1	\$20.08	\$20.08
Standard-Wall Unthreaded PVC Pipe for Water -5ft.	1	\$5.40	\$5.40
90 Degree Elbow Connector, White, 1 Socket-Connect Female	1	\$0.61	\$0.61
PVC Cement Primer - 8oz.	1	\$6.07	\$6.07
PVC Pipe Cement - 8oz.	1	\$5.27	\$5.27
Dual Wall Adhesive Marine Heat Shrink Kit - 3:1 Shrink Ratio - Black	1	\$14.97	\$14.97
Conical Storage Tank	1	\$87.33	\$87.33
6mm Flexible DC Motor Hex Coupler	1	\$3.46	\$3.46
12V 35RPM Worm Gear Motor	1	\$23.88	\$23.88
L298N Motor Drive Controller Board Module	1	\$6.89	\$6.89
45 degree - PVC Pipe size 1, 45 degree connector Female	3	\$0.99	\$3.99
1/8", 3' long steel rod	1	\$2.99	\$2.99
2" PVC cap	1	\$2.58	\$2.58
2" PVC coupling	1	\$0.98	\$0.98
2" PVC Bushing	1	\$1.48	\$1.48
M4 Hex Bolts	1	\$1.65	\$1.65
M3 bolts	1	\$2.15	\$2.15
1-1/4" to 1" PVC adapter	1	\$2.58	\$2.58
3/8" PVC, 5' long	1	\$2.48	\$2.48
5/16" box of rubber washers	1	\$5.98	\$5.98
150-300GPH Smartpond fountain pump	1	\$39.57	\$39.57
1/2" plastic tubing (for fountain pump)	1	\$6.88	\$6.88
box of Hex bolts (used for metal base)	1	\$6.00	\$6.00
box of nuts for hex bolts used for metal base	1	\$8.00	\$8.00
Hillman 8-ft x 1.25-in Plated Steel Slotted Angle - used for metal base	2	\$24.98	\$49.96
3" x 1.5" steel sloted angle - 14 inches long	5	-	\$0.00
IP66 Electrical Control box	1	\$34.70	\$34.70
Waterproof Toggle Switch	1	\$22.95	\$22.95
Arduino	1	\$20.69	\$20.69
plastic lining (trial for cone in hopper)	1	\$4.77	\$4.77
Dispensing Piece	1	\$ -	\$0.00
Flap to dispense fish	1	\$ -	\$0.00
Fish			
		TOTAL	\$434.33

Figure 20: Table - Build of Materials

Part of Project	# of Hours	Total Price (@\$30/hour)
Planning	45	\$1,350
Construction & Testing	95	\$2,850

Figure 21: Table - Estimated Labor Costs

Timeline

The timeline for the first half of the semester (09/04/18 - 12/05/18) was focused on research, planning and preliminary documentation for the project. The second half (01/16/19- 04/19/19) of the semester was then heavy on implementation and testing of the design. A more detailed timeline can be seen in the Gantt Chart in Figure 22 below.

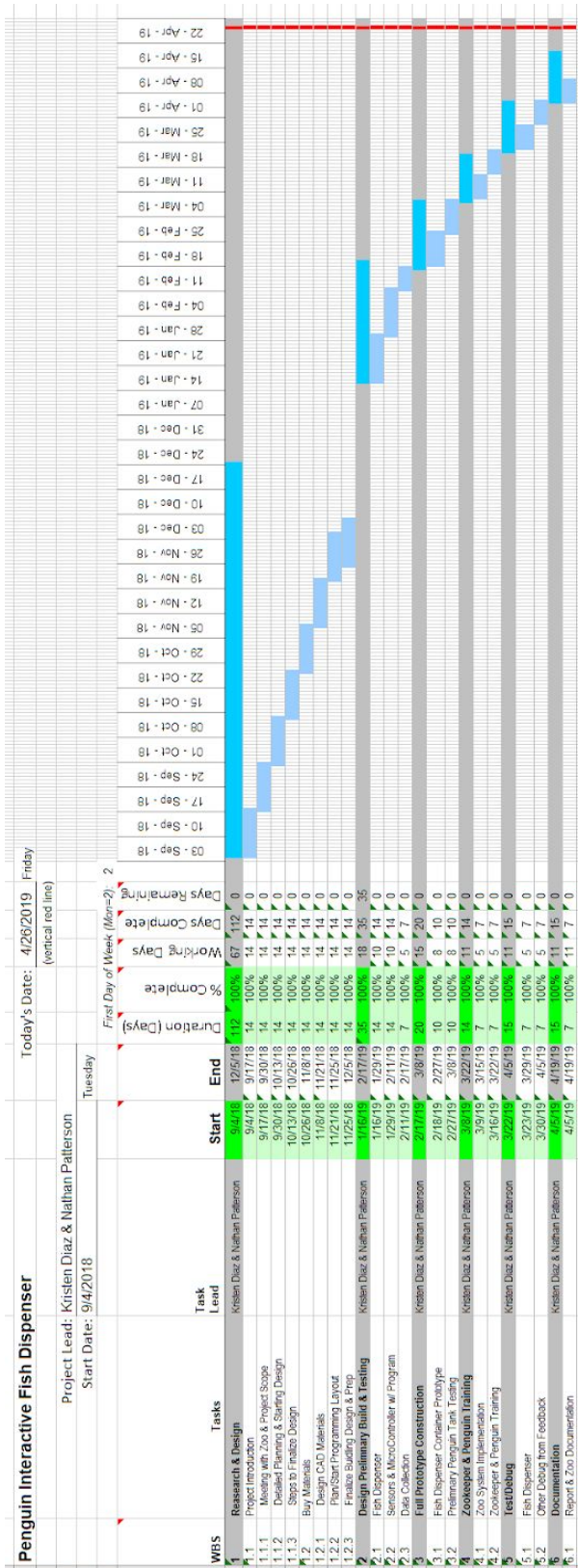


Figure 22: Gantt Chart Timeline

Problems Encountered/Analysis of Problems Solved

Penguins Being Able to Press a Button

In the initial designs, we had a push-button in place of the toggle switch. However after further discussions with the zoo on the skills of the penguins, a toggle switch design was brainstormed, and a replacement waterproof toggle switch was replaced into the system design.

Fish Flow ‘Mechanics’

A possible issue known from the beginning was the concept of how fish would be able to flow through the system, especially through the hopper without getting clogged. Multiple designs of various pipe materials, types, and sizes were tested and a water pump was added to help ease the flow of fish through the system. Fish were also cut down into more rectangular pieces to try and cut down the amount fish pieces would be able to bend during travel. There was also an issue with larger plastic pipes purchased that were zip-tied so tight in its packaging that the circular shape of the tubes that were to be used were permanently squeezed into an oval shape, permanently deforming the best shape for fish travel.

However, all of the tests of other design options proved insufficient, limiting the overall fish load to only a few fishes at a time before fish began to get clogged at some point in the hopper or tubing. Unfortunately due to the limited time with the end of the semester, we did not have enough time to re-haul the fish-holding design, to come up with an entirely different design to test that could work for fish flow.

Future Recommendations

Due to the overwhelming issues that came about with the flow of fish through the system - something that does not even exist on the commercial market from what we know. Future recommendations would be to have a team to heavily focus on just the fish flow mechanical aspect of the project. So possibly senior design students with mechanical engineering backgrounds that could collaborate with our technical advisor, Dr. Dieter Vanderelst.

One possible solution a mechanical engineering professor suggested at the Senior Design Expo, was if it would be possible to design a fish holding system where the fish would be constantly agitated by water (circulated in the holding area) until needed by the dispensing system, by which time it could be selected/sucked out by the system to be dispensed. Possibly eliminating a chance for multiple fish to get stuck together and clog the system.

The electrical design and components will be able to be re-used and picked up by the new team if they wish to use them. And the Arduino code is commented and simple enough for mechanical engineering students to be able to pick-up with the help of Dr. Dieter Vanderelst.

Then, once the Interactive Penguin Fish Dispenser can handle a load of dead fish at a time, the system can then be fully implemented at the Cincinnati Zoo. If the outdoor temperature low is above 60°F, the temperature where the little blue penguins can be kept outside, then the zoo keepers will be able to easily train the penguins by food-reward training. Once the penguins are trained to hit the modified toggle switch, the system can be officially installed in the exhibit by mounting the brackets into the ground on the edge of the exhibit's pool.

CONCLUSION

The final outcome of the Interactive Penguin Fish Dispenser was not ideal. However, it was a good experience with a lot of new terrain to explore, such as 3D modeling, working with an input that was very biological and unpredictable (dead fish & fish mechanics), and learning about many many different types and sizes of tubing. We also learned a lot from Dr. Katie Kalafut & Rickey Kinley about how little blue-foot penguins operate which was very different from what we normally learn in engineering fields. We hope that a team that is much more mechanically experienced can finish up this project with the Cincinnati Zoo so that the blue penguins will finally have a reliable and automatic interactive experience that improves their enrichment in their exhibit.



Figure 23: Picture of two of the little-blue-foot penguins at the Cincinnati Zoo

REFERENCES

1. “Arduino DC Motor Control Tutorial - L298N | PWM | H-Bridge.” *HowToMechatronics*, 8 Feb. 2019, howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/.
 - a. This contained information on how to integrate the L298N Motor Driver with the Arduino and DC motor in order to have directional and speed control of the motor. We used the information to set up our motor driver correctly and used some of the commands to control the motor for our final code.
2. “Arduino Forum - Index.” *Arduino Forum - Index*, forum.arduino.cc/.
 - a. This forum was used multiple times to help configure various components of the circuit. The forum contains a large variety of information regarding different coding problems with the Arduino boards. It helped with controlling the DC motor as well as switching out the starting push-button with a toggle switch.
3. “Button.” *Arduino*, www.arduino.cc/en/tutorial/button.
 - a. This website shows how to properly set up a push-button on the breadboard and connect it to the Arduino. This was helpful in the early stages of the design because the push-button first used to control the motor.
4. Erlacher-Reid, C, et al. “Evaluation of Potential Variables Contributing to the Development and Duration of Plantar Lesions in a Population of Aquarium-Maintained African Penguins (*Spheniscus Demersus*).” *Zoo Biology*, U.S. National Library of Medicine, May 2012, www.ncbi.nlm.nih.gov/pubmed/21557300.
 - a. This article describes why Bumblefoot disease may be happening more frequently with birds held in captivity than in the wild. This contributes to why the project is necessary to help limit these factors and hopefully decrease the frequency of Bumblefoot.
5. “How to Spin a DC Motor with the Arduino.” *Dummies*, www.dummies.com/computers/arduino/how-to-spin-a-dc-motor-with-the-arduino/.
 - a. This contained information on how to control a DC motor with the Arduino. It was used to inform us of the correct set up of the components on the breadboard before we got the separate motor driver. It also helped us with the different commands to use in order to get better control of the motor.
6. L. Reisfeld, M. Barbirato, L. Ippolito, R. C. Cardoso, M. Nichi, M. G. Sgai, and C. S. Pizzutto, “Reducing bumblefoot lesions in a group of captive Magellanic penguins (*Spheniscus magellanicus*) with the use of environmental enrichment,” *Pesquisa Veterinária Brasileira*, vol. 33, no. 6, pp. 791–795, 2013.

7. Luiza, et al. "Study of Bacteria Isolated from the Foot Pad of Spheniscus Magellanicus with and without Bumblefoot." *Semantic Scholar*, 1 Jan. 2013, www.semanticscholar.org/paper/Study-of-bacteria-isolated-from-the-foot-pad-of-and-bact%20C3%A9rias-coxins/4ca59ea3326ba34fa9fd936de0c1ba2c8416e2d1.
 - a. This was used to inform on the purpose of the project, of why Bumblefoot is an issue for penguins at the Cincinnati Zoo and how water enrichment could help reduce the issues cause by the Bumblefoot disease.
8. Oracle, DIY. "DIY Automatic Pet Feeder Made From Water Pipe." *Hackster.io*, 16 Mar. 2018, www.hackster.io/diy-oracle/diy-automatic-pet-feeder-made-from-water-pipe-a10fde.
 - a. This website shows how to make an automatic pet feeder for dogs and cats. It gave us the initial idea for how to dispense the fish on our device. We got the idea to 3D print the flap inside the tube to stop and allow the fish to pass through. We also got the initial idea to use the Node MCU for our project but later changed it to an Arduino Uno board.
9. "StateChangeDetection." *Arduino*, www.arduino.cc/en/Tutorial/StateChangeDetection.
 - a. This website had information on how to detect a change of state with the push-button on the Arduino. We used this because we wanted to be able to turn the motor once while the button/switch was activated and stop even if it was still in the same position. We then wanted to be able to turn the motor again when the button/switch was activated in the different position.

APPENDIX A: Arduino Code

```
//The motor will go down roughly 90 degrees
//and then return to the original position
//This occurs when the switch is pressed in either direction

//Motor driver pins
int enA = 10;
int in1 = 6;
int in2 = 7;

//Toggle Switch pins
static const int switchPin = 5;

//Initializing switch state
int switchState = LOW;
int lastSwitchState = LOW;

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);

  pinMode(switchPin, INPUT);
  lastSwitchState = digitalRead(switchPin);
}

void loop() {
  switchState = digitalRead(switchPin);
  if (lastSwitchState != switchState) {

    if (digitalRead(switchPin) == HIGH && switchState == HIGH) {
      //make motor go down
    }
  }
}
```

```

digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
analogWrite(enA, 75);
delay(1000);

//make motor go up
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
analogWrite(enA, 75);
delay(1000);

//make motor stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
delay(2000);
}

else if (digitalRead(switchPin) == LOW && switchState == LOW){
//make motor go down
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
analogWrite(enA, 75);
delay(1000);

//make motor go up
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
analogWrite(enA, 75);
delay(1000);

//make motor stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);

```

```
    delay(2000);  
  }  
  delay(2000);  
  lastSwitchState = switchState;  
}  
}
```

APPENDIX B: COMPONENT MANUALS

STMicroelectronics - L7809CV (9V Voltage Regulator)



L78

Datasheet

Positive voltage regulator ICs



Features

- Output current up to 1.5 A
- Output voltages of 5; 6; 8; 8.5; 9; 12; 15; 18; 24 V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection
- 2 % output voltage tolerance (A version)
- Guaranteed in extended temperature range (A version)

Description

The L78 series of three-terminal positive regulators is available in TO-220, TO-220FP, D²PAK and DPAK packages and several fixed output voltages, making it useful in a wide range of applications.

These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type embeds internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

Maturity status link

L78

2 Pin configuration

Figure 3. Pin connections (top view)

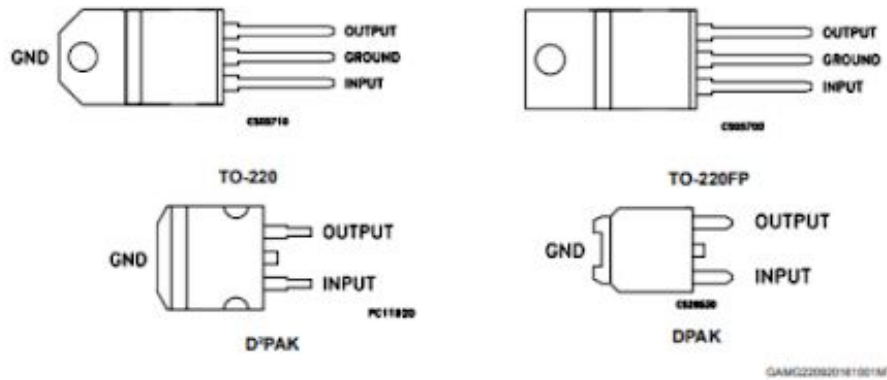
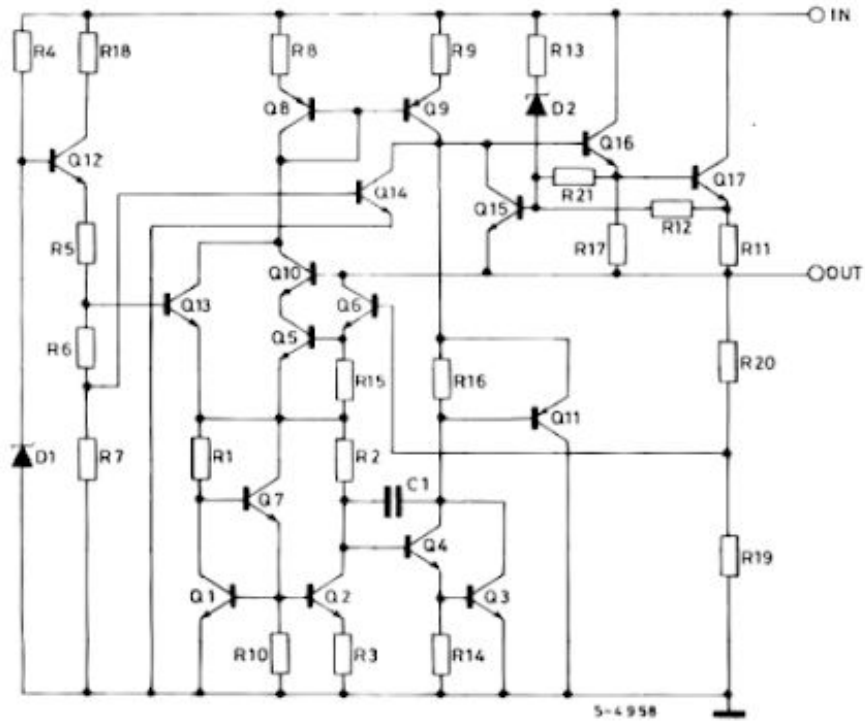


Figure 4. Schematic diagram



3 Maximum ratings

Table 1. Absolute maximum ratings

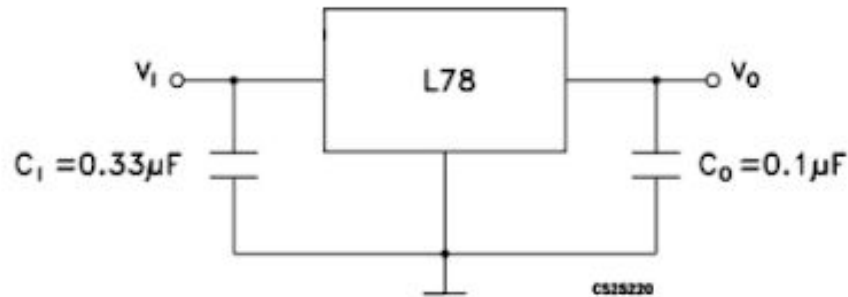
Symbol	Parameter		Value	Unit
V_I	DC input voltage	for $V_O = 5$ to 18 V	35	V
		for $V_O = 20, 24$ V	40	
I_O	Output current		Internally limited	
P_D	Power dissipation		Internally limited	
T_{STG}	Storage temperature range		-65 to 150	°C
T_{OP}	Operating junction temperature range	for L78xxC, L78xxAC	0 to 125	°C
		for L78xxAB	-40 to 125	

Note: Absolute maximum ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 2. Thermal data

Symbol	Parameter	D'PAK	DPAK	TO-220	TO-220FP	Unit
$R_{\theta JC}$	Thermal resistance junction-case	3	8	5	5	°C/W
$R_{\theta JA}$	Thermal resistance junction-ambient	62.5	100	50	60	°C/W

Figure 5. Application circuits



GAK22062019-1003MT

Arduino UNO

Arduino Uno



Arduino Uno R3 Front

Arduino Uno R3 Back



Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

[Revision 2](#) of the Uno board has a resistor pulling the BU2 HWB line to ground, making it easier to put into [DFU mode](#).

[Revision 3](#) of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Toggle Switch

Specifications:
 Single Pole Single Throw
 On-Off
 30A 12VDC
 20A 125VDC
 10A 277VDC
 1½HP
 Mounts in a .48" diameter hole and has 6 in long wire leads
 UL and CSA approved

SIZE	WP28A	REV.
A		0
REVISIONS		
LET.	WAS	DATE
0	Original Release	

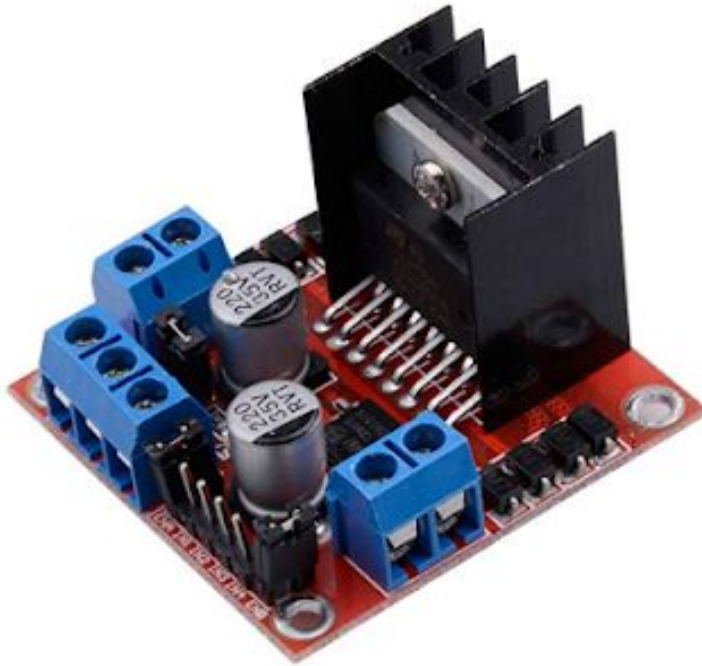
Mounting Hole

Circuit Diagram

DECIMAL TOLERANCES UNLESS OTHERWISE SPECIFIED: DIMS .0005 HOLES .001 DIA .003 ANG .015 R/A .01/32 X .01/16 ANGULAR .01" SURF .01" X .01"	All dimensions in inches (unless otherwise noted) GAMA Electronics, Inc. P.O. Box 1488, Crystal Lake, IL 60039	THIRD ANGLE PROJECTION
TITLE Toggle Switch		
INFORMATION CONTAINED HEREIN IS PROPRIETARY INFORMATION OF GAMA ELECTRONICS INC. THE REPRODUCTION, USE OR DUPLICATION OF THIS INFORMATION FOR ANY PURPOSE OTHER THAN THAT FOR WHICH THE INFORMATION IS PROVIDED, IS PROHIBITED, EXCEPT BY EXPRESS WRITTEN PERMISSION.	DATE DRAWN IN COMPLIANCE WITH ISO-9001-1987 RELEASED	MATERIAL DIR: KEG 7/16/14 SHEET 1 OF 1

L298N Motor Driver

Qunqi L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper For Arduino



- **Specification:**
- Chip: L298N
- Logic voltage: 5V
- Logic current 0mA-36mA
- Storage Temperature: -20 °C to °C to +135
- Operating mode: H-bridge driver (dual)
- Drive voltage: 5V-35V
- Drive current: 2A (MAX single bridge)
- Maximum power: 25W
- Dimensions: 43x43x27mm

Note:



- This module has a built-in 5v power supply, when the driving voltage is 7v-35v, this supply is suitable for power supply
- DO NOT input voltage to +5v supply interface, however leding out 5v for external use is available.
- When ENA enable IN1 IN2 control OUT1 OUT2
- When ENB enable IN3 IN4 control OUT3 OUT4