FPL INNOVATORS

Final Detailed Design

Nicholas Grant, Andrew Lois, SirDarius Lomack, Nicholas Haynes, Jacob Ray, Christian Perez

FAMU-FSU College of Engineering

EML 4911: Senior Design

Jerris Hooker & Shayne McConomy

March 22, 2024

Final Detailed Design

Design Concepts:

Team 304's project had two different avenues to solve the issue that FPL provided either improving the current method of replacing a blown fuse which includes the extended hook stick device, or a remote switching device that would be placed on the utility pole. After weighing out our options on what would be more effective on reducing the strain on the linemen we went towards the remote switching device because of the implication that it will reduce the amount of times the linemen would have to travel to the worksite.

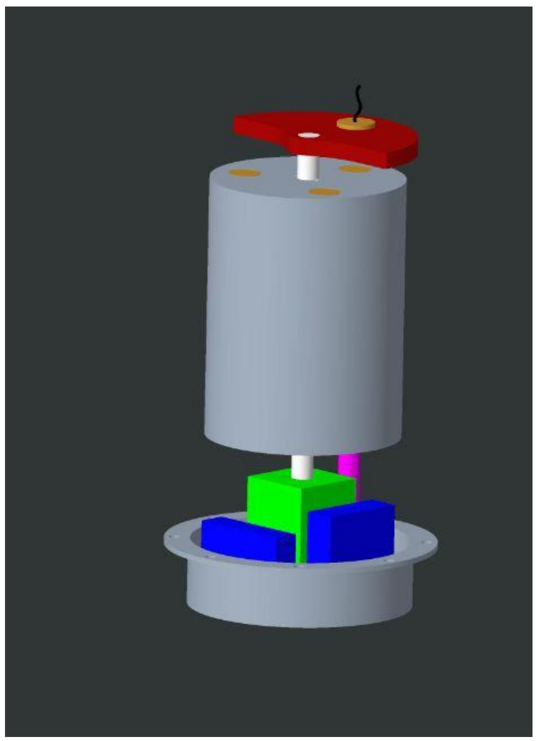


Figure 1: First design concept

This graphic is from the very first iteration of our team's design. One of the biggest challenges that our group faced with proceeding with a developed version of this idea was not being able to redesign the fuse itself. Florida Power and Light has millions of dollars invested in fuses for their cutout switches and it would be foolish to switch to a different fuse system and allow those to be wasted. Another aspect of this design that required changing was the motor and the microcontrollers were located in a case in the bottom of the device. We decided to move away from this because it would be easier to ensure there is no collection of water in the motor's compartment if it is isolated at the top.

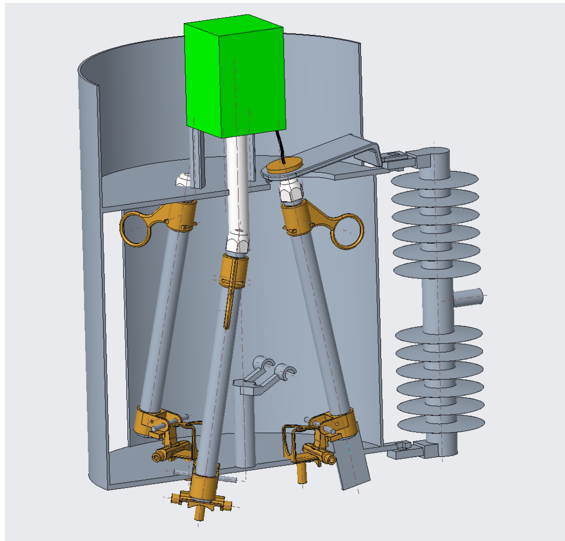


Figure 2: Preliminary detailed design for the revolver

Our first detailed design of the revolver implemented the changes from our first idea, with the current fuses being used and the motor being isolated on top, and included a contact point for the bottom of the fuse that was on the same axis of the revolver barrel. The main issue with this design was the process of moving the fuse into a position where it could be engaged. Once the fuse is burnt and is disengaged, how would the motor turn the next fuse into the position where it can be pushed up and engaged. Our solution for this problem was to create a housing that holds the fuses and rotates with the motor's shaft.

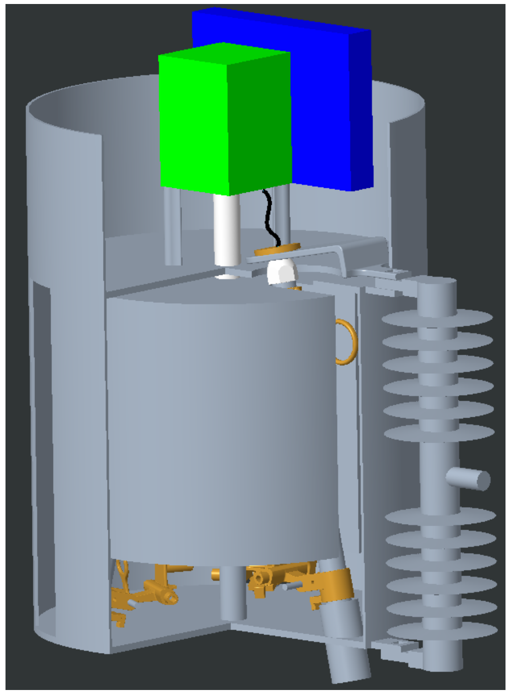


Figure 3: Detailed design with the fuse casing implemented

This design implemented the housing for the fuses that could allow for them to be turned into position once engaged. From this design, we realized we had numerous issues. The first of which was that there would be no way for the fuses to be swapped out without disassembling much of the device. There needed to be a way for the fuses to be accessed from outside for the device to be worth using on the power grid. The second issue was that the fuse needed to be rotated so that the contact point for the bottom could be moved away from the middle of the device. The contact point being connected to the shaft presented some complications with the energization of the system, if all of the connected parts have current flowing through them it would pose a major safety hazard. The third issue was that the bottom of the fuses was not supported, so we needed to design a track system that the fuses can sit in while being turned into the position to be engaged.

Final Detailed Design:

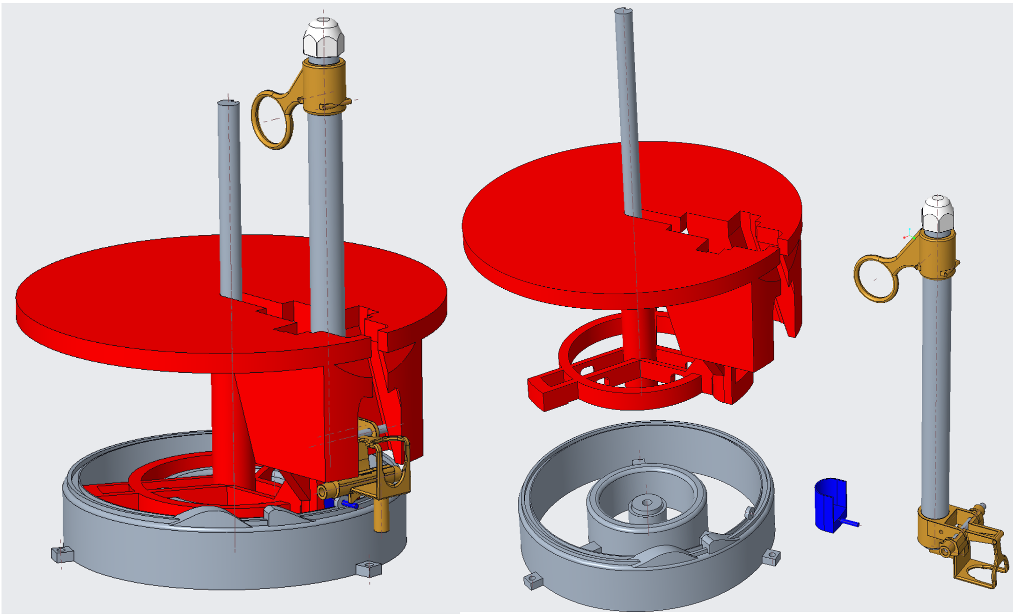
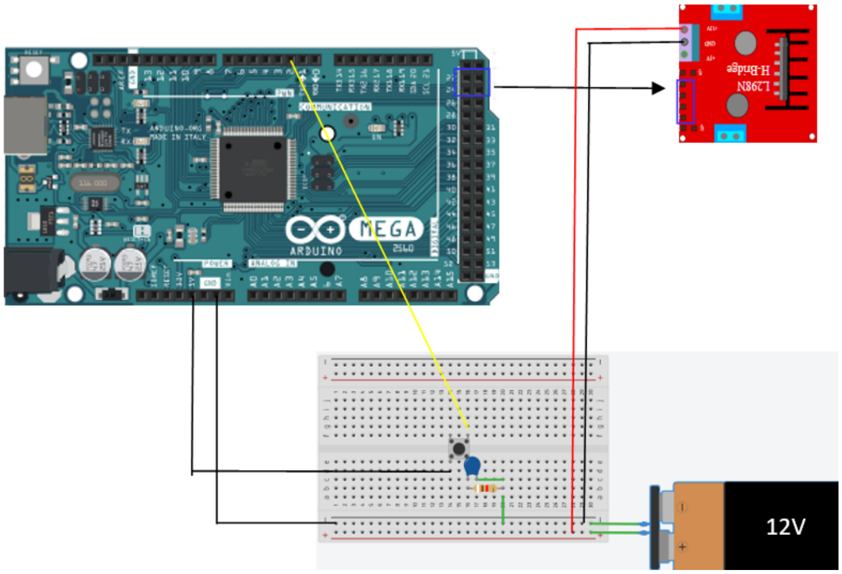


Figure 4: Final remote switching device

The final design of our device includes 3 main components the foundation track, the rotating catch, and the fuse holder. The foundation remains fixed and provides a path for the rotating catch to follow. The rotating catch features a tapered access point that matches the bottom shape of the fuse to make it easier for the linemen as it will fall into place with ease. Once placed in the device the user can trigger the switching device via a button to rotate the fuse over the track where it will raise 0.5 inches and drop into the contact points. Once the fuse is triggered the mechanical spring that keeps the hinge open will release cause the fuse to drop 1.5 inches where it can be rotated out and replaced.

Electrical Circuit:



We have a push button on the breadboard, connected to the 5V supply of the arduino, a resistor and capacitor in parallel with each other, and pin 2 on the arduino's pulse width module. The resistor and capacitor are then connected to ground. When the button is pressed, pin 2 will read as high, prompting signals to be sent to the motor driver to make the motor spin. These signals are sent from digital pins 22-25 of the arduino to the motor driver. The Motor driver is also connected to the positive and ground buses of the breadboard, which is supplied by our 12V battery.

Code:

The pseudocode used for this assignment is as follows:

Initialize variables

buttonPin = Pin 2

LED Pin = Pin 13

Void Setup

Port A = output

Pin 2 = input

Pin 13 = output

Void Loop

Read buttonPin

switch(state)

Case 0:

    if(button=HIGH && (counter % 3) != 0)

        Drive motor 67 steps

        Turn test light on

    Else if((button=HIGH && (counter % 3) = 0)

        Drive motor 66 steps

        Turn test light on

    Else

        Light is off, no movement

Case 1:

    Completed 67 steps

    Return to case 0

Case 2:

    Completed 66 steps

    Return to case 0

Always return to Case 0 whenever anything is completed

Actual Code:

```
//initialize variables

int currentPosition = 0;
int desiredPosition = 0;
int counter = 1;
int state = 0;
int buttonState = 0;          //sets up state of the putton to 0
```

```
const int buttonPin = 2;       //sets the button pin to 2
const int ledPin = 13;         //sets LED pin to pin 13


//declare step order
int fullSteps[4] = {0b1, 0b1000, 0b10, 0b100};


void setup() {
  DDRA = 0xFF; //using port A as output
  pinMode(buttonPin, INPUT);  //takes input from the button pin
  pinMode(ledPin, OUTPUT);    //sets ledpin to output
}

void loop() {
  buttonState = digitalRead(buttonPin);
  switch(state) {
    case 0:
      if((buttonState == HIGH) && ((counter % 3) != 0)){  //if button is
pressed and counter is not divisible by 3, enter loop
        currentPosition = 0;
        desiredPosition = 67;         //set desired position to 67 to
drive motor
        digitalWrite(ledPin, HIGH);   //checks to see if button input is
read
        counter++;                    //iterate counter
        state = 1;                    //move to state 1

        }
      else if((buttonState == HIGH) && ((counter % 3) == 0))  {   //if
button counter is divisible by 3, rotate 66 steps
        currentPosition = 0;
        desiredPosition = 66;          //set desired position to 66 to
drive motor
        digitalWrite(ledPin, HIGH);    //checks to see if button input is
read
        counter++;                     //iterate counter
        state = 2;                     //move to state 2

      }
```

```
      else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    }

    break;


    case 1:
      if(currentPosition == 67){  //reset state to inert (0) after
rotating 67 steps
        state = 0;
      }
    break;

    case 2:
      if(currentPosition == 66){   //reset state to inert (0) after
rotating 66 steps
        state = 0;
      }
    break;
  }

if(currentPosition < desiredPosition){  //drive motor to desired postion
  currentPosition++;                    //if motor is not at the desired
position, move forward
  PORTA = fullSteps[currentPosition % 4];
  delay(30);
  state=0;
}
else if(currentPosition = desiredPosition){

  state=0;
}
else{

  state=0;
}
}
```