A Temperature Controlled Catapult with LED Warning

Nicole Malinowski

CS 207

December 6, 2019

**Table of Contents**

Introduction and Background 3

Design Process 3

Catapult Building Process 5

Setbacks and Failures 6

User Manual 7

Mile Stones 8

Future Improvements 9

Conclusion 9

References 10

Appendix 11

Appendix A Catapult Photos 11

Appendix B Schematic Details 12

**Introduction and Background**

My project for this semester was a catapult that would launch if the temperature was above a predetermined threshold. I also have a color changing LED that acts as a warning light as the temperature increases. I decided on this project because sometimes I don't realize how warm my room gets when I have my space heater running. So now I can have the catapult to remind me to turn my space heater off.

The set-up is fairly basic. A temperature sensor takes data in and compares it to preloaded thresholds in the code. Depending on where the temperature is in relation to the thresholds it can change the colour of the RGB LED and potentially causes the catapult to launch. The catapult is built from K’nex and an elastic band for tension. This is an adjusted design of Wasagi’s K’nex catapult. A servo arm holds the catapult down and when the threshold is reached the servo arm moves releasing the catapult. There is a button that can be pressed to reset the servo arm to hold down the catapult arm again without the need to restart the program.

The idea for the temperature sensor catapult came from Ryan Jones’ pressure sensor catapult. This was easily adjusted to be controlled by a temperature sensor instead of a pressure sensor as well as adding a RGB LED.

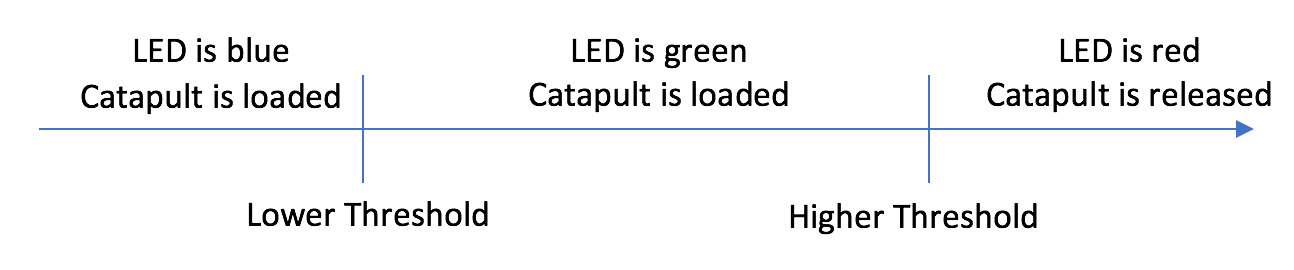
**Design Process**

The code for Ryan Jones’ pressure sensor catapult is fairly basic. In set up, the servo arm was set to be holding the catapult arm down. Then it took input in via the pressure sensor and compared it to a threshold. If it was larger than the threshold the servo arm would move launching the catapult. When the button was pressed it would move the servo arm back into position to hold the catapult down.

Adjusting the code to use a temperature sensor instead of a pressure sensor was fairly easy. The temperature sensor was wired to use the same input pin the pressure sensor would have used. However, the threshold needed to be adjusted to reflect the current ranges that would reasonably be read in. Working on my project at a later stage I converted the input from the temperature sensor from volts to the corresponding degrees Celsius so setting the thresholds would be more user friendly.

I added a RBG LED which acts as a warning light. The light is originally set to blue, the cool setting. As the temperature increases the LED changes to green, and eventually when the temperature gets too warm and the catapult launches the LED changes to red. After the catapult has launched, the catapult will need to be reset manually by pulling the catapult arm down and pressing the button for the servo arm to move. However, the LED will continue to monitor the temperature and adjust colours reflecting the current temperature. This allows the LED to represent the current temperature and the catapult to show whether or not the temperature has gotten above the threshold at any point since the catapult has been reset.

The LED and servo are all controlled within if statements. Since there are three different states there needs to be two different thresholds. The lower of the two thresholds controls when the LED will change from blue to green. The higher threshold controls when the LED changes from green to red and the catapult launches. Within the if statements the colours of the RGB LED are adjusted and with the higher threshold the servo arm also moves. The button which resets the servo to hold the catapult arm down is also controlled in an if statement. A diagram of the thresholds and states of the LED and catapult is shown on the next page. Stability of the LED around the threshold points is discussed in the Set Backs and Failures section.



**Catapult Building Process**

The K’nex catapult build was based off of Wasagi’s tutorial on Instructables. His tutorial has a rectangular base with an upright triangle on two opposing sides of the base. The tops of the triangles are connected with a K’nex piece on which clips hold an elastic band. The elastic is then also around the basket of the catapult which allows for tension on the elastic band when the catapult is lowered towards the base. This set up was fairly good but needed some adjustments.

There were two main adjustments that were needed to adapt the original set up to be compatible with the rest of the project. The original catapult set up did not have anything to hold the catapult arm down once you moved your hand away. Therefore, a piece needed to be added into the interior of the base that would secure the servo allowing it to hold the catapult arm down. The servo is secured to the K’nex piece with elastic bands. I have not had any trouble with the servo staying in place so the elastic bands are enough to secure the servo. To load the catapult after it has been released, the catapult arm needs to manually pulled down and the button on the Arduino needs to be pressed to move the servo arm back into the locked position.

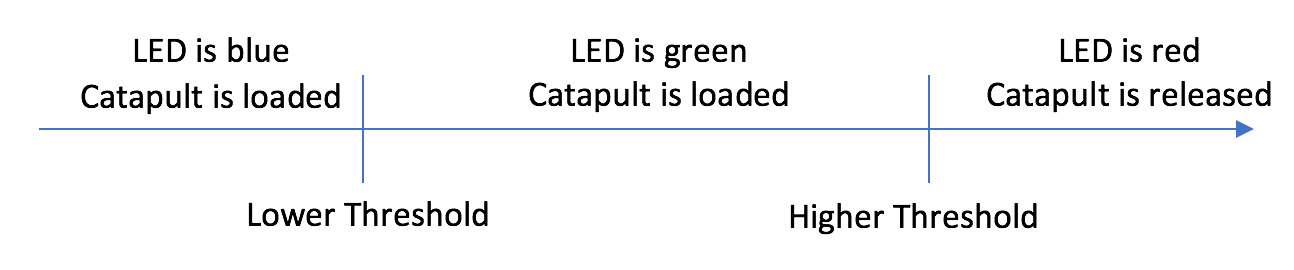
The second main adjustment was due to the catapult arm swinging out from under the servo arm while the servo arm remained in the loaded position. This was because the tension from the rubber band was pulling the servo arm slightly upwards. This was then allowing the catapult arm to slide upwards and out from underneath the servo arm. The combat this I added a piece onto the base that has two upright pieces that the catapult arm slots into while it is in the loaded position. This forces the catapult arm to stay in line under the servo arm as it is no longer able to slide sideways and out from underneath. See Appendix A for pictures of the constructed catapult.

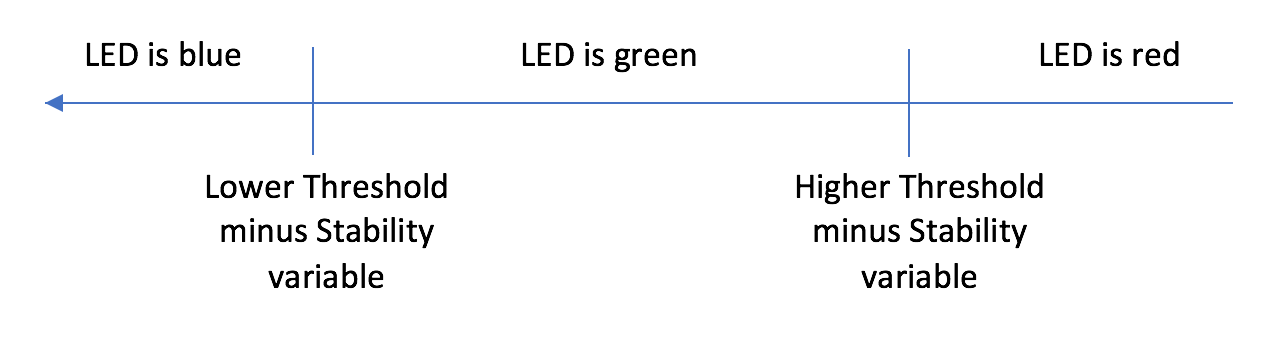
**Setbacks and Failures**

While working on my project there were some setbacks and problems that arose. One of them was the catapult arm swinging out from under the servo arm that was previously discussed. Along with this engineering problem there was also a hardware and software problem that arose.

The hardware problem was being created when the button was pressed to reset the servo arm into the loaded position. Pressing the button was causing voltage to flow backwards and cause the reading on the temperature sensor to spike upwards thus causing it to read over the threshold to release the catapult which then moved the servo arm back to the released position. Adding a fly back diode between the voltage source and the button ensured that pressing the button would not influence the readings taken from the temperature sensor.

The software problem encountered was when the temperature sensor was reading values jumping just above and just below the threshold. This was causing the LED to rapidly change colours as the readings flip above and below the threshold. This caused the LED to act like a coloured strobe light. To fix this, I added in a stability variable that is designed to be set to a low positive value. I have it set to 0.2 but anywhere between 0.1 and 0.5 would be reasonable. It also should be smaller than the difference between to upper and lower threshold. It works by changing the thresholds when the temperature is decreasing to be, in this case 0.2 degrees, lower than the thresholds while the temperature is increasing. For example, if the lower threshold was set to 20 degrees and the higher threshold was set to 22 degrees with a stability variable of 0.2 then when the temperature is increasing the LED would change colours at 20 degrees and 22 degrees. However, when the temperature was decreasing the LED would change colours at 21.8 degrees and 19.8 degrees. This causes the LED to be more stable because if the it needs at minimum a 0.2-degree change in temperature to affect the LED. Whereas, before a very small change in the temperature could cause the LED to change colours and bounce back and forth between the colours as the temperature fluctuates. The diagram below illustrates this with the arrow showing the direction of the temperature change.





**User Manual**

Setting up and using the temperature controlled catapult is not difficult. The parts needed are fairly basic and include: a servo motor, RGB LED, temperature sensor, push button, diode, resister, 17 wires, and some building materials for catapult. As mentioned earlier, I used K’nex and an elastic band for the catapult and it worked very well. See appendix A for photos of the K’nex catapult and Appendix B for the schematic and breadboard diagram circuit set up.

The circuit set up is fairly straight forward. There is a button wired to input pin two, a temperature sensor wired to analog pin zero, and a servo motor wired to input pin nine. As well as a RGB LED with the blue leg wired to pin thirteen, green leg wired to pin twelve, and red leg wired to pin eleven.

The user may need to change the values for three variables in the code depending on what they want for thresholds and stability. These variables are all in the top block of code. The lower threshold that changes the LED from blue to green is called blue2green. The higher threshold that causes the LED to change from green to red and the catapult to launch is called green2launch. The stability variable that was described in the previous section is simply called stability.

**Mile Stones**

Looking back now that it is the end of the semester, the project undertaken was not super ambitious. However, considering at the start of the semester I wouldn’t have even been able to wire a simple LED this project seemed gigantic when I started working on it.

The basics for my project went fairly smoothly. I’m glad I was able to find a tutorial for a basic K’nex catapult that I could just modify for my project. This made the building process way shorter than I was expecting. As well, adjusting Ryan Jones’ code to use a temperature sensor instead of a pressure sensor was much easier than I anticipated. It was just a matter of figuring out the wiring and changing the thresholds to an appropriate level. After the catapult was built and the code was adjusted I had extras to add onto my project.

I originally had planned to add an LED to the project that would light up when the catapult was launched. However, since the project had been going well I decided to opt for a colour changing LED. Adding in a colour changing LED rather than just a normal LED required additional coding because I had two different thresholds for the state of the LED. Whereas if I just had an LED light up when the catapult was launched it would have just been adding code into the pre-existing if statement.

**Future Improvements**

To add onto this project in the future I would like to see if I could add a motor that would manually wind down the catapult arm when the button is pressed. The motor would have a string tied to it that is also attached to the catapult arm. Then, when the button is pressed the motor would run for some predetermined amount of time then shut off. The servo arm would then move to hold the catapult in place. Then afterwards the motor would run in the opposite direction to unwind the string so it does not interfere with the next catapult launch. Trying to calibrate the motor to determine how much time the motor should run for to lower the catapult arm back under the servo arm would require some trial and error. As well, the motor would need to run in both directions so it would require an H bridge.

**Conclusion**

Overall, I am very happy with the way my project turned out. It is not a super advanced project but it is way more than I would have thought I would have been able to do before I started the class. Seeing the finished result and being able to work through problems and find a solution was very satisfying. This was one of my only university projects that was very hands on and I was able to actually physically see the result of my work. Being able to put something together and have it function the way I intended it to is super cool.

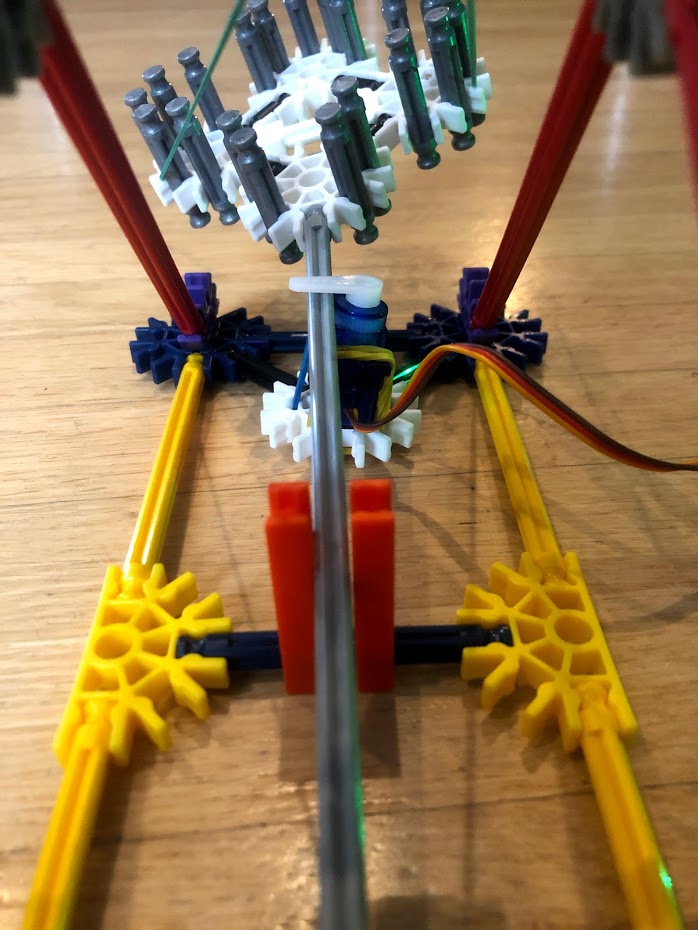
**References**

Wasagi. “Small Knex Catapult.” Instructables, Instructables, 10 Nov. 2017, www.instructables.com/id/Small-Knex-Catapult/.

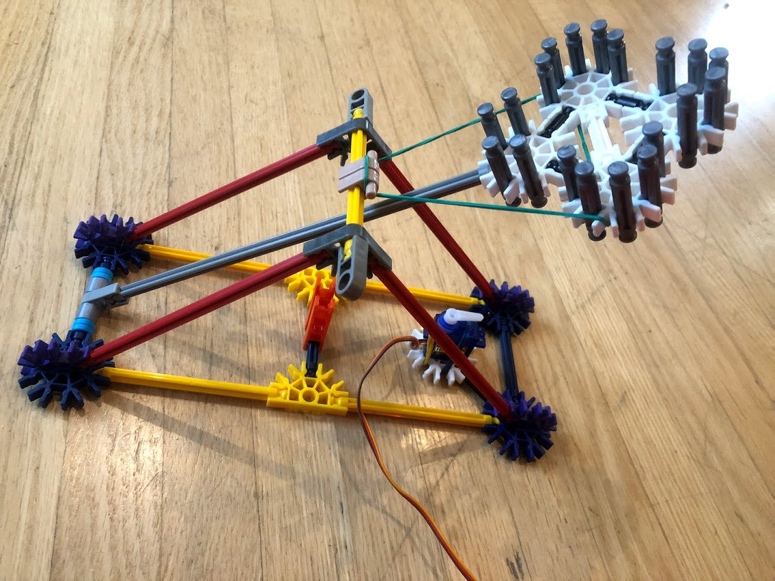
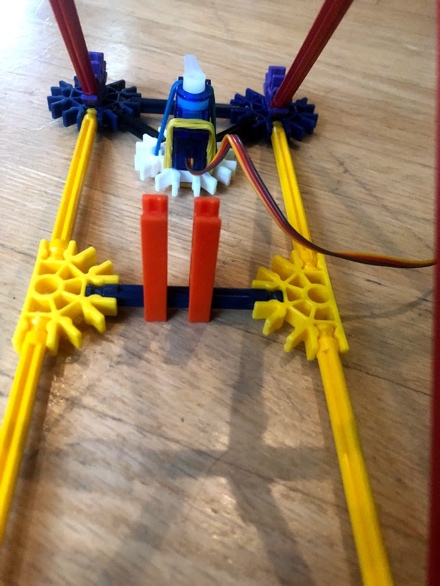
Jones, Ryan. “How to Make a DIY Catapult With Arduino: Arduino.” Maker Pro, Maker Pro, 18 Mar. 2017, maker.pro/arduino/projects/arduino-servo-catapult.

**Appendix**

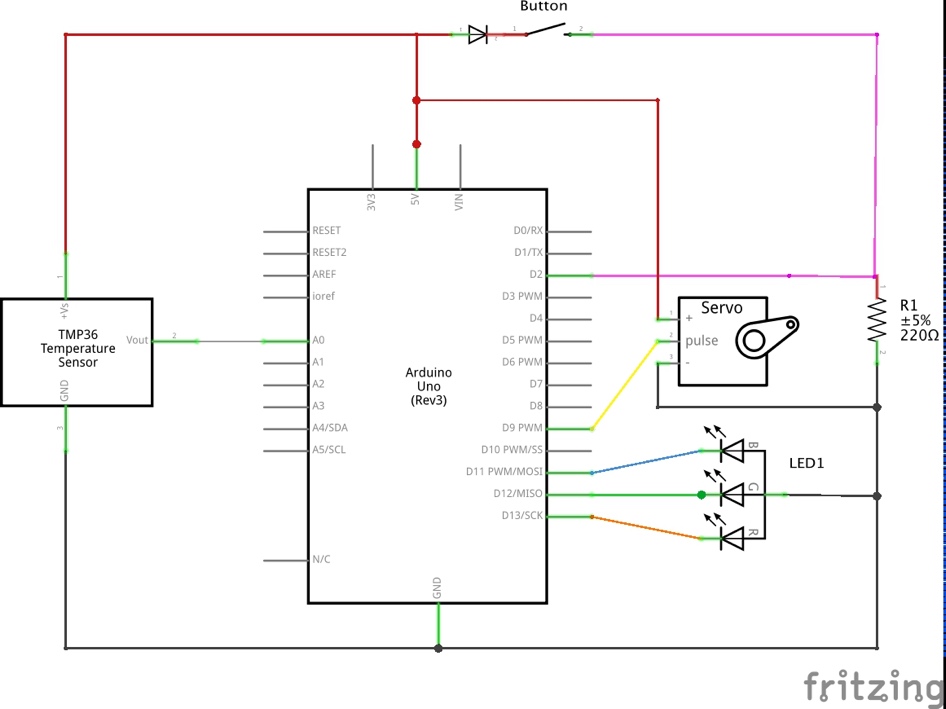
**Appendix A Catapult Photos**

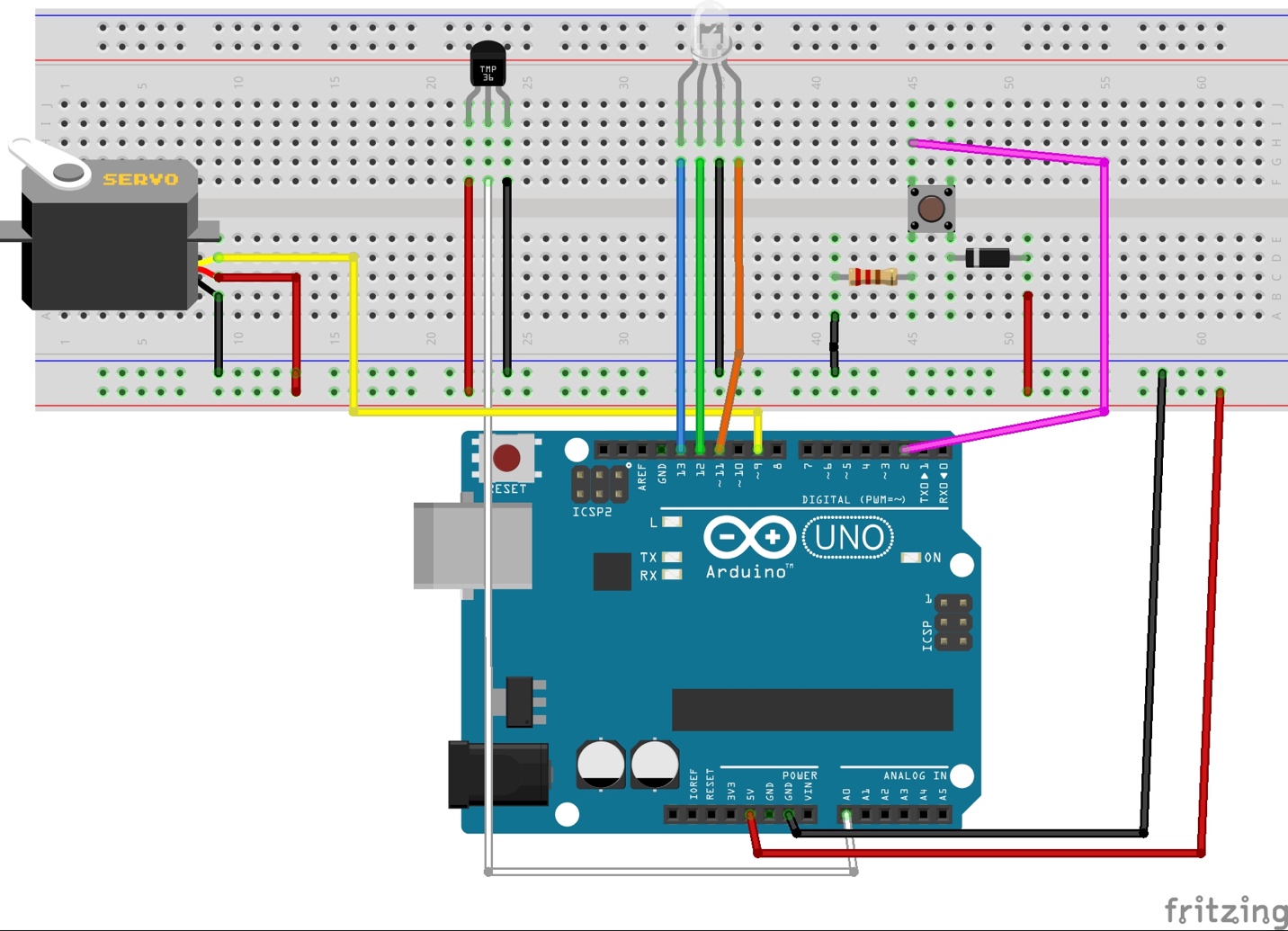
Photos of the catapult in the loaded position. Note the two orange pieces holding the catapult arm in place so it doesn’t like out from underneath the servo arm holding it down.

Photos of the catapult in the released position. Note the servo arm has now moved and the tension on the elastic bands caused the catapult arm to move.

**Appendix B Schematic Details**

****

****