**Project Overview**

The goal of this project was to design a safety system for a piece of laboratory equipment, specifically the fume hood in the Mina lab. Over time the maximum safe operational height for the sash of the fume hood lowers. While there was a system to prevent the sash from going above its original maximum safe operational height, there was no system in place to stop the sash from going above its maximum safe height as said height adjusted over time. The safety system would have an adjustable height and would alert users when the sash passed a certain height, meaning it would enforce the maximum safe operational height even as the height changed over time.

The first step in the process of designing the system was determining the type of sensor I would be using to detect when the fume hood passed a certain height. I thought of three main types of sensors that I thought would work best, which were as follows:

* A motion detector, which would visually detect the end of the sash passing in front of it
* A photoresistor and some form of a light source such that the sash raising would interrupt a beam of light travelling to the photoresistor
* A hall effect sensor, which would detect a magnet attached to the fume hood passing by

I ordered each type of sensor and tested each of them out, for their various strengths and weaknesses:

* Motion detector
  + Pros:
    - Small and compact, would lead to a simple circuit design
  + Cons:
    - Very sensitive, may detect non-sash movement
    - Would sense movement behind the glass of the sash
* Photoresistor
  + Pros:
    - Very reliable, could consistently detect sash passing by
    - Wouldn’t be triggered by anything other than sash
  + Cons:
    - Requires extra LED
    - This would lead to a more complicated and larger circuit
* Hall effect sensor
  + Pros:
    - Very small, extremely compact circuit design
    - Wouldn’t detect non-sash movement
  + Cons:
    - Fairly unreliable detection

From these results, it was clear to me that the photoresistor was best as, while it would lead to a more bulky and complex circuit design, it was the only sensor that was reliable and would not lead to false detections. As such I moved forwards with designing the circuit with the photoresistor detection circuit.

There were three main components to the final circuit, the detection system, the processor, and the alert system. The detection system would consist of a photoresistor with a constantly lit LED positioned directly opposite. One leg of the photoresistor would be connected to the ground and the other would be connected to an input pin on the Arduino nano I would use as a processor, as well as a pull-up resistor pulling the pin high. When the light was fully shining on the photoresistor from the LED it would have a high resistance and as such the pull-up resistor would pull the input to the Arduino’s input pin high. However, if something were to come between the LED and the photoresistor, blocking the light, the resistance of the photoresistor would drop, and since one of its legs was connected to the ground it would overcome the pull-up resistor and pull the input pin low. As such, we could expect a consistently high pin, except when the LED was interrupted, which would pull the pin low. The Arduino would then detect this and send out a signal to the alarm system, bringing a pin high which would power both an LED and a small buzzer-type speaker to alert the user.

Once I had this plan I got to work implementing it on a protoboard which went well. The only slight problem I hadn’t originally accounted for was that the LEDs needed a resistor on them to prevent burnout, but this was easy to implement. I ended up with a protoboard with the Arduino at one end, with its micro USB port on the edge for easy access, and the photoresistor and LED hanging off the side of the protoboard. This allowed space for something attached to the sash to slide along the side of the protoboard and interrupt the LED’s light, triggering the alarm.

I next got to work writing the code, just a simple system to check at set intervals the value it was reading on the sensor systems input pin. If it ever reads a low value, it sends a signal for a set period of time to the alarm system before turning it off and having a short cooldown period, to prevent repeat detections. It would then go back to actively sampling the sensor system. The code I have provided also contains the code that would be used for both the motion detector and hall effect sensor.

Once this was done I took measurements of the fume hood and protoboard and got to work designing housing for the project which would enclose the board, fit onto the fume hood, and allow space for a piece connected to the sash of the fume hood to block the photoresistor. My final design featured a simple box to enclose the protoboard, with rails and a screw hole to help support and mount the board inside the enclosure. On the underside, the side attached to the fume hood, it had pieces to fit the geometry of the side of the fume hood and small holes for magnets to fit into, which would secure the enclosure to the fumehood. It also has a slot running vertically along the length of the side facing the sash, with holes on either side of the slot, one for the photoresistor and the other for the LED. A thin piece of plastic attached to the sash could then move through the slot as the sash moved up and down. Then, if the sash passed a certain height, it would block the LED, triggering the alarm system. The enclosure also featured a hole on the bottom for the power cord to connect to the Arduino, holes on the top for the alarm LED and speaker, and a small invented line on the top to show the maximum fume hood height.

I printed out this enclosure and, while the first attempt had some errors in some of the dimensions, the second version worked perfectly. The magnets held the enclosure securely to the fumehood and the protoboard was held securely in the enclosure, with the LED and photoresistor facing each other from across the slot. Then, after designing a small piece which fit onto the bottom of the fume hoods sash which would pass through the slot, which was simply a small piece of plastic that was held in place with friction, the system was fully operational.

The system is easily adjustable to detect the fume hood passing any maximum height which is set. It also consistently detects the fume hood raising above a certain height and alerts its users and has had no false detections since it has been turned on. It also has no negative impact on the operation of the fume hood.

Overall I would say that the process of designing this fume hood has been very successful and the final product operates exactly as intended and fixes the problem it was designed to solve.

**Ideas For Improvement**

While the final product of this project does behave as intended, there are still some ways it can be improved. One of the main areas for improvement would be the overall size of the final product. The final sensor with its casing is rather large and bulky and could be reduced in size significantly. Most of the final size of the product is due to the use of a protoboard to help with the wiring. If this protoboard was not used and instead the components were soldered directly to the Arduino nano board with wires then the overall size of the product could be reduced significantly to only a fraction of its original size. The added flexibility in the design due to the lack of a protoboard could also help with other improvements as it would mean components could be repositioned much more easily. For example, in addition to reducing the overall size, you would be able to do things like create a deeper slot down the side is the photoresistor is more easily triggered.

There are also some other simple improvements that can be made. For example, the port for the power cable at the bottom is not entirely centred on the charging port and is off by about a millimetre. This is a simple error in the design and while it does not impact use it is still something which can be improved.

The most drastic improvement that could be made is the complete removal of the Arduino nano. This would require a complete redesign of the circuit but I believe it is possible t design a circuit with all the same functionality of what the Arduino nano provides in the original design. This would not only save on space but lso save significantly on power consumption, and mean that the design could potentially be powered by batteries rather than a power cable which would make it much more convenient to use.