SUPERCOMPUTING CHALLENGE

ROBOTIC WILDFIRE DETECTION SYSTEM: SMOKEY

Capital High School

Team 09 members:

Zachariah Burch

Isel I Aragon M

Britny Marquez

Daniel Leon Leon

Teacher Sponsor:

Barbara Teterycz

Mentor:

David Ritter

Table of Contents

Abstract/Executive Summary

The purpose of this project is to address and help with the frequent wildfires that are happening in the state of New Mexico every year. Regardless of whether wildfires are controlled or uncontrolled, they should always be treated very seriously and with extreme caution because they are the source of pollution, and are life threatening. By releasing CO_2 into the atmosphere, wildfires also contribute to global warming. Furthermore, forests purify the air that we breathe, offer a home to much of the world's wildlife and flora, and provide natural resources (e.g. timber and medical plants).

The result of the project is a robotic fire weather conditions detection system called Smokey, which alerts about either very possible or already existing fires. It includes the following parts: Arduino MEGA 2564, CO_2 and TVOC sensor, Dust and Pollen sensor, Temperature and Humidity sensor, RTC Clock, BlueTooth and SD drives, infrared light sensor, parabolic dish, LCD display, and several LEDs. Smokey was programmed using C++ language and has been tested during different experiments and through the data collection and analysis. The experiments included burning incense, candles, and matches, turning humidifiers on/off, and exposing Smokey to an infrared heater.

Smokey correctly detects increased/decreased levels of CO_2, TVOC, smoke, dust, temperature, and humidity in the air, as well as it correctly senses IR (infrared) light reflected in a parabolic dish to the IR detector caused by the heat of the flames, which were burnt in the chemical fume hood. This fire weather conditions detector could be used by the Fire Services to determine when to avoid prescribed fires. In addition, by installing a network of such robots in the forest that would communicate with each other via Bluetooth and be powered by the PV solar energy, the Forest Services would be quickly alerted about the wildfires that have already started on their own.

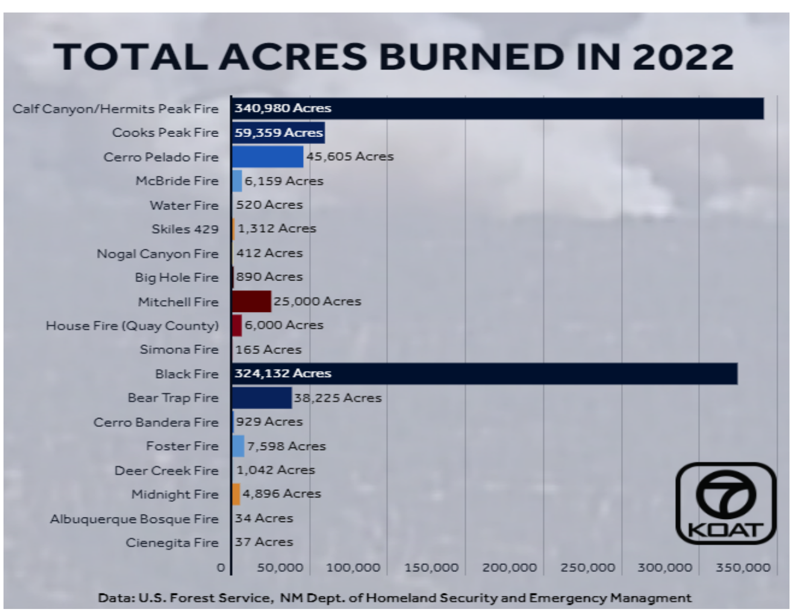
Hypothesis

High levels of dust, smoke, CO_2, and temperature, and low level of humidity in the air as well as the presence of infrared light of a flame detected from far away may indicate possible wildfires and wildfire weather conditions. Developing a robotic system with sensors that would measure all these levels and detect such an infrared light could inform about already existing wildfires or predict the potential ones.

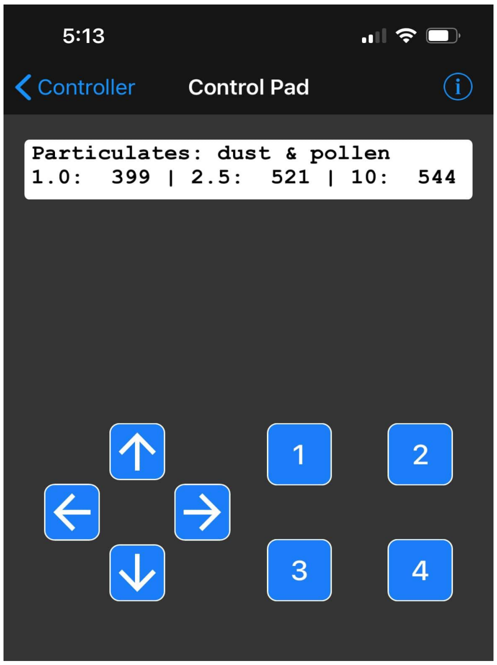
Identify the Problem

Problem Background and Research

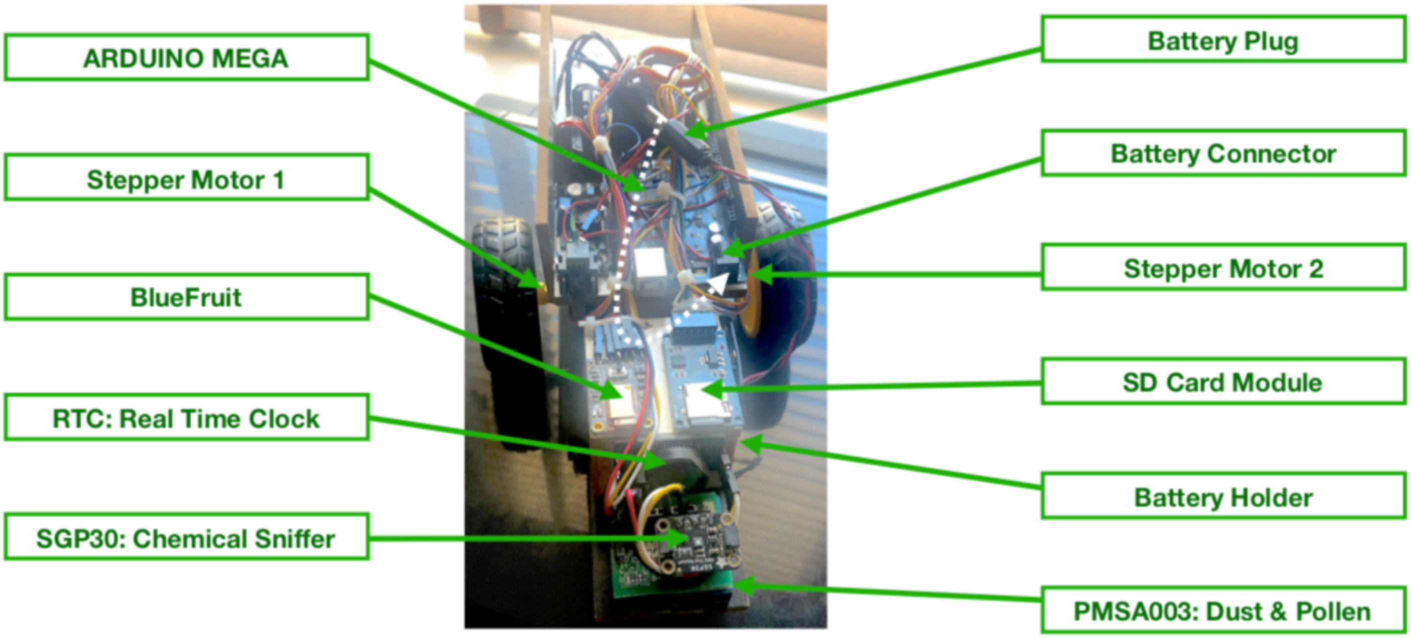
According to InciWeb, “the Hermit's Peak fire began April 6, 2022 as a result of the Las Dispensas prescribed fire on the Pecos/Las Vegas Ranger District of the Santa Fe National Forest” (InciWeb). While “the Calf Canyon fire was caused by a pile burn holdover from January that remained dormant under the surface through three winter snow events before reemerging in April” (InciWeb). As the fires reemerged and grew, they then managed to merge and become one, known as The Hermit's Peak and Calf Canyon fire. All together more than 800,000 acres burned in the state of New Mexico in 2022.



The fires that emerged became immense to the point that the smoke was visible in Santa Fe. Since the team had already developed an air quality monitoring system called Snoopy (that was submitted to the Supercomputing Challenge last year), they were able to measure the level of dust and smoke in the air during that time. The images below show those levels as well as the smoke captured here, in Santa Fe.



Previous Project: Snoopy



Idea Generation / Goal

As mentioned before, Snoopy was able to detect the level of dust, smoke, pollen, TVOC (Total Volatile Organic Compounds), and CO2 in the air. Based on the collected data during the time of wildfire, the team was inspired to upgrade Snoopy to a fire and fire weather measuring system. This is where the idea of Smokey came from.

Work Done By Others

In order to implement this idea and accomplish this goal, the team was regularly meeting with the mentor from industry and their teacher. During the meetings the team learned about each part of hardware, including laying out circuit boards and adding sensors to it, and C++ coding.

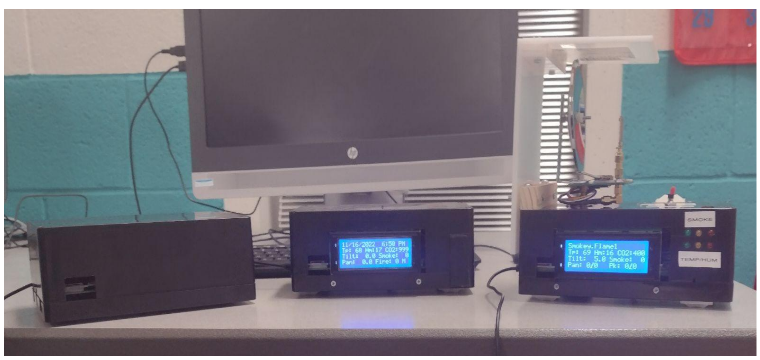
Constraints

Since some of the team members were unable to meet regularly, after meeting with the mentor, the team leader was updating the rest of the team after school. It was a very challenging and time consuming role to do both work with the professional mentor and teach the rest of the team.

Model/Prototype

Description of Model

Smokey is a robotic monitoring system that detects the levels of dust, smoke, CO_2, TVOC, humidity, temperature, and infrared light.



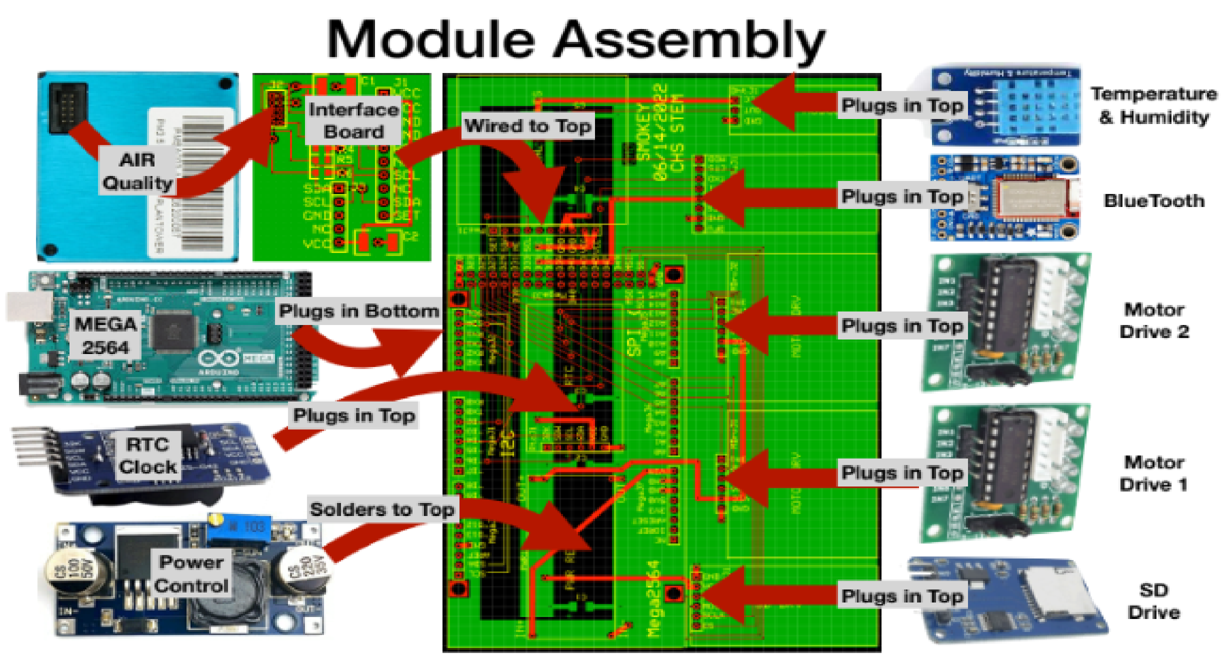
The development of Smokey included the following three phases displayed on the image above:

1. Protype #1 (on the left) only included the basic sensors;
2. Prototype #2 (in the middle) in addition to basic sensors, also included warning LEDs and LCD (Liquid Crystal Display);
3. Prototype #3 (on the right) in addition to components in prototype #2, also included infrared sensor and parabolic dish.

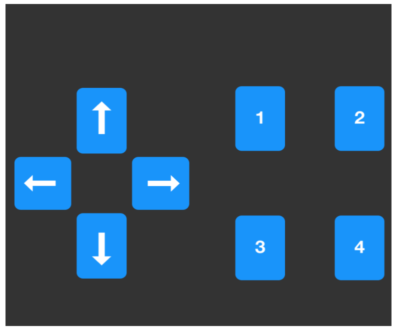
Smokey reused sensors and their software used by Snoopy that detected smoke, dust / pollen particles, as well as TVOC and CO2. The following parts were also added to Smokey: circuit board, power module, infrared, humidity and temperature sensors. The table below lists similarities and differences between Snoopy and Smokey.

Snoopy	Both	Smokey
• Uses wires to connect the sensors • Uses stepper motors to drive around • Detects air quality	• Have CO2, TVOC, pollen sensors, RTC, and SD card modules. • Use C++ coding	• Uses a circuit board with a modular interface • Has an IR sensor, a humidity and temperature sensor, and a power module. • Uses stepper motors to pan and tilt IR sensor and dish • Detects wildfires

As shown on the image below, in order to build Smokey, the following hardware was required: Arduino Mega, circuit board, power control and bluetooth modules, SD and motor drivers, RTC clock, air quality, and temperature and humidity sensors.



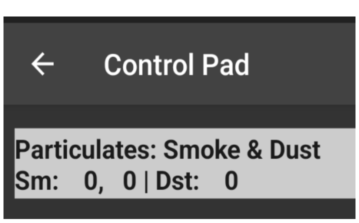
The team learned how to connect Smokey to the Bluefruit app installed on their smartphones through Bluetooth, and how to control the infrared sensor and parabolic dish movement when each of the following arrow buttons was pressed:

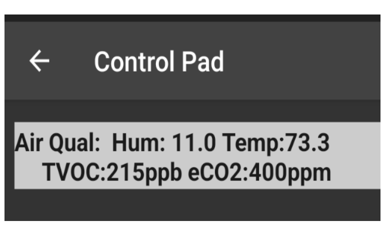


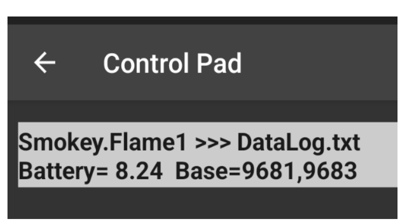
In order to accomplish this task, the team had to add the following instructions to the code:

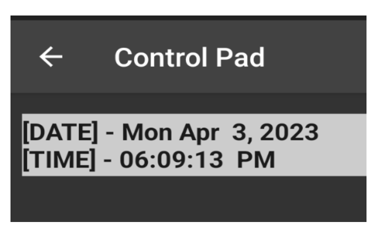
```
  newCmd = GetCommand();        // Always read from the App input on bluetooth
  if(newCmd < 0 || newCmd >8) { // Illegal command - cancel it
    currentCmd = 0;             // Zero out the command variables
    newCmd =0;
    MotorsOff();                // Make sure motors are off
    }
  ActiveCmd = (currentCmd<=8) && (currentCmd>=5); // 5<=Cmd<=8 is active command
  if(ActiveCmd == 0)  currentCmd = newCmd; // If its not active cmd, update it
// Active commands implemented
  if((currentCmd == 8) && (Auto == 0))  Pan(1);      // Pan right
  else if((currentCmd == 7) && (Auto == 0)) Pan(-1);   // Pan Left
  else if((currentCmd == 6) && (Auto == 0)) TiltDown(1);  // Tilt Down - not used at present
  else if((currentCmd == 5) && (Auto == 0)) TiltUp(1);    // Tilt Up - not used at present
```

In addition, the team also learned how to program each of the four buttons to display data on the screen of the Bluefruit's app. Button 1, when pressed, displays the level of smoke and dust in the air; button 2 displays the level of humidity, temperature, and chemicals, such as CO2 and TVOC, button 3 displays the level of battery, and button 4, when pressed together with button 1, displays the date and time, as shown on the screenshots below:







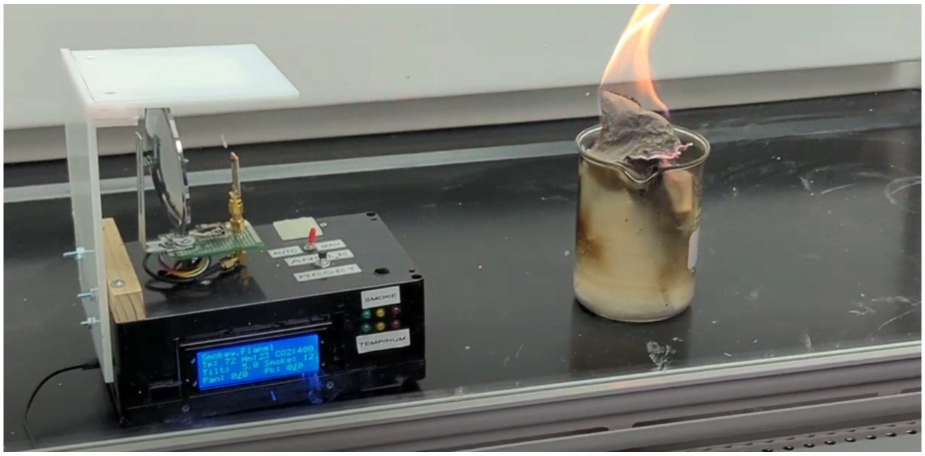


The screenshot below shows the part of the code responsible for accomplishing the above tasks.

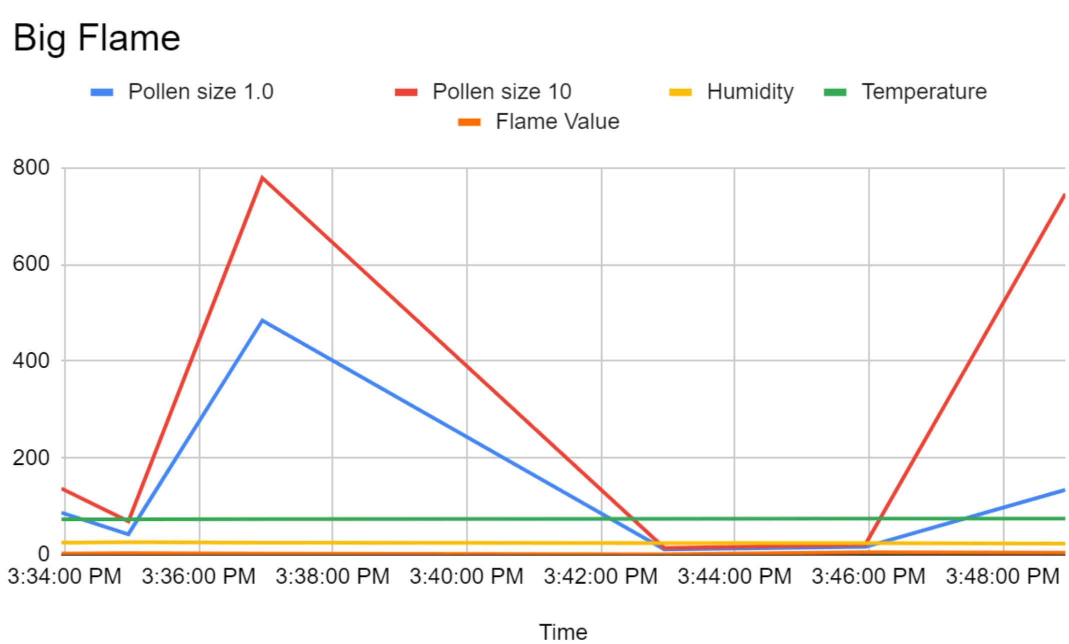
```
  else if(currentCmd == 4){          // Pressed button 4 ... Date and Time
    AppShift = TRUE;
  } else if(currentCmd == 3){    // Pressed button 3 ... Battery & baselines
    if(AppShift != FALSE) {      // For App Display #n6
      currentDisplay = 6;
      updateFlm1Dsply();
      updateDisplay();
    } else {               // For App Display #3
    currentDisplay = 3; // Each button goes through same sequence
    updateNameDsply();
    delay(100);
    updateDisplay();
    currentCmd = 0;       // Commands 4,3,2,1 are button presses and get cancelled
    newCmd =0;            //   once implemented.
    delay(100);
    }
  } else if(currentCmd == 2){    // Pressed button 2 ... Chemical data CO2 and TVOC
    if(AppShift != FALSE){       // For App Display #4 & # 1
      currentDisplay = 5;
      updateFlm0Dsply();
      updateDisplay();
    } else {
      currentDisplay = 2;
      updateAirQualDsply();
      delay(100);
      updateDisplay();
      currentCmd = 0;       // Commands 4,3,2,1 are button presses and get cancelled
      newCmd =0;            //   once implemented.
      delay(100);
    }
  } else if(currentCmd == 1){    // Pressed button 1 ... Particulate data
    if(AppShift != FALSE){       // For App Display #4
      currentDisplay = 4;        // Button number is same as display number - stopped here
      updateTimeDateDsply();     // Format Time/Date for display
      delay(100);                // Give it some time to settle
      updateDisplay();           // Write out to display
      currentCmd = 0;      // Commands 4,3,2,1 are button presses and get cancelled
      newCmd =0;           //   once implemented.
      delay(100);          // Wait for settling again
    } else {                     // For App Display #1
      currentDisplay = 1;
      updatePartDsply();
      delay(100);
      updateDisplay();
      currentCmd = 0;       // Commands 4,3,2,1 are button presses and get cancelled
      newCmd =0;            //   once implemented.
      delay(100);
```

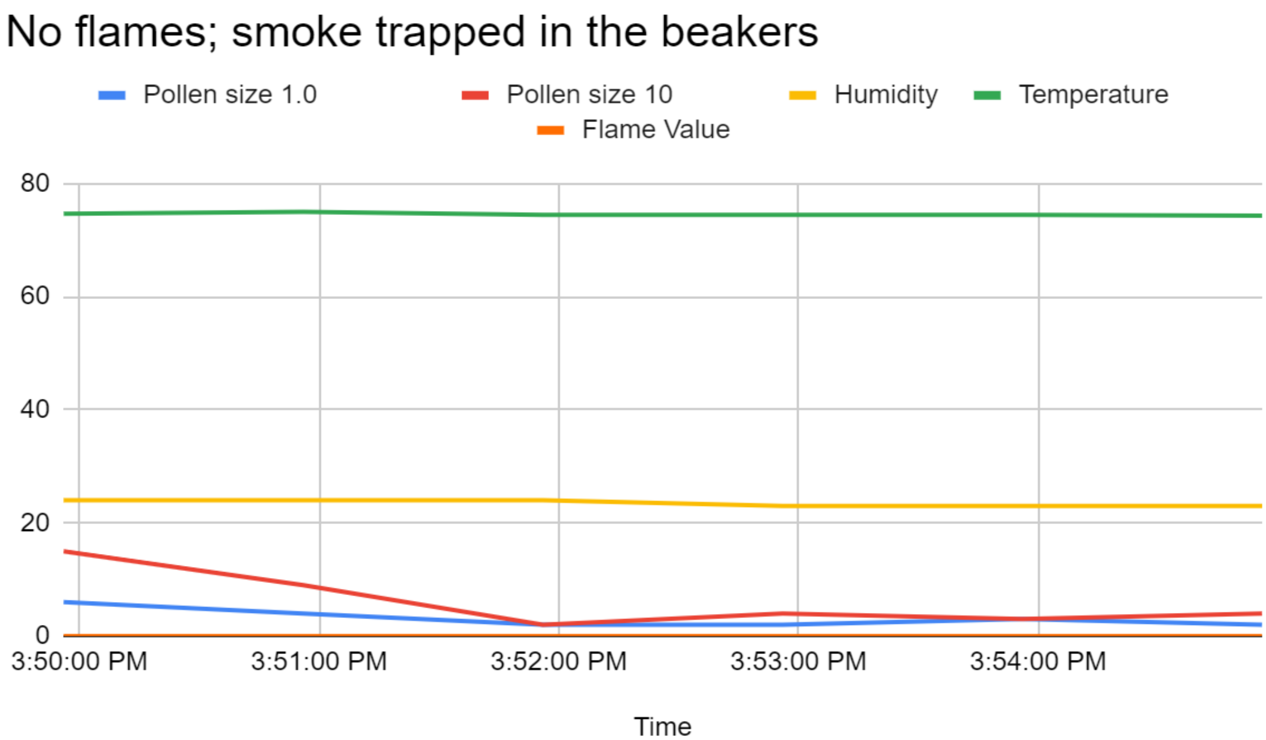
Test model and evaluate

Testing

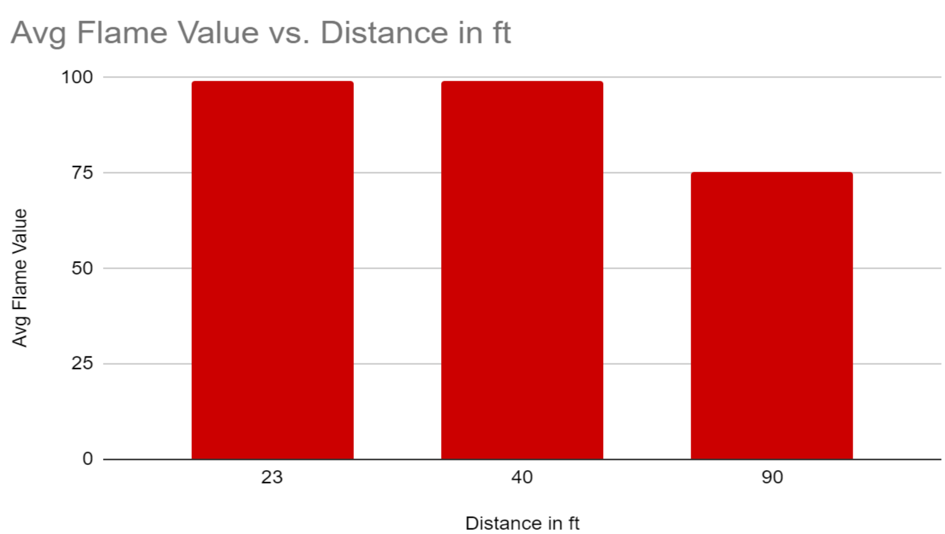


Many tests were performed with Smokey, including exposing it to the burning flames in the chemical fume hood. The performance of Smokey was recorded and saved onto the SD card and then analyzed and presented graphically in the charts below. Each line within the charts represents a value that was recorded by Smokey, the blue line represents the pollen size 1.0 values, the red line represents the pollen size 10 values, the yellow line represents the humidity values, the green line represents the temperature values and the orange line represents the flame values of this experiment.





In addition to the actual flames, Smokey was also used to detect light from an infrared heater. The chart below displays the efficiency of the range of the infrared sensor from different distances. It was able to easily get a reading all the way up to 90 feet. The experiments were performed in the classroom as well as school corridors. This is proportional to the inverse square law. Every time the distance was doubled, the efficiency of the infrared sensor dropped to ¼. Due to the limitations of the number of characters on LCD display and the Bluefruit app used on cell phones, the maximum value of detected infrared light displayed by Smokey is 99.



Troubleshooting

Before and after the many tests, bug fixes were added to the C++ code and libraries of Smokey. The occasional bugs were observed in the values displayed on the LCD panel. One initial bug was due to an error caused by the fluctuation of voltage in the faulty wiring within the battery pack. Other bug fixes included sorting and separating the code into three different sections that made it more readable and easier to troubleshoot. Another bug that occurred within the LEDs was caused by the pin values being flipped for the smoke, humidity, and temperature LEDs. A recurring issue was with the CO2 module as it would not update to display the correct value. As of this time, most of these bugs have been resolved.

Redesigning

As previously mentioned, Smokey has been redesigned several times as improvements were made to it, leading to three different prototypes. Each next prototype was an improved version of the previous one with some extra components added to it. The idea for adding an infrared sensor to the last prototype of Smokey came from the satellites that use a parabolic dish. As for Smokey, it was built around the size of its circuit board and components. This was done in order to have enough space to fit everything within it and to have the ability to mount it in various places, including the actual forest, buildings, and homes.

Computational/Mathematical Model

The code consists of more than a thousand lines split into 3 different libraries. The first library calls upon the setup process for Smokey's modules and Smokey's DO functions and starts a watchdog timer. The watchdog timer keeps Smokey's software from freezing for long periods of time as it counts out each second. After a series of 10 seconds, a flag is set and the watchdog timer counts to a minute and sets another flag. If the software is still frozen after 5 minutes, the software resets. This unfreezes Smokey's software, and the data still stays saved onto the SD card.

```
ISR(TIMER5_OVF_vect){   // interrupt service routine every half second
  TCNT5 = timer5_counter;   // preload timer (from arduino app notes)
  Flag05sec = TRUE;
  if(timer5_repeat == 0) {  // need to skip every other call because
      timer5_repeat =1;     //  this timer runs every half second.
    } else {                // Only come here once per second.
      timer5_repeat = 0;    // Clear "repeat" flag.
      Flag1sec = TRUE;      // Set the 1sec flag.
      Count10sec = (Count10sec + 1)%10;       // Update 10 sec counter
      if (Count10sec == 0) Flag10sec = TRUE;  // Set 10sec flag on overflow
      Count1min = (Count1min +1)%60;          // Update 1min counter
      if (Count1min ==0) Flag1min = TRUE;     // Set 1min flag on overflow
      CountDisplay = (CountDisplay +1)%DisplayTime; // Update Dispaly counter
      if(CountDisplay == 0) FlagDisplay = TRUE;     // Set display on overflow
      WatchDog += 1;                          // Update watchdog counter
      if(WatchDog >= WatchDogDly) {           // If watchdog hits delay, reset processor
                                              // delay set to 10 at start, up to 300 at setup
        Serial.println("WDog Rst in 1 sec");  // Print watchdog warning
        delay(1000);                          // Pause for printint
        resetFunc(); // if WatchDog not reset in loop for 5min then reboot machine.
      }
    }
```

The second library is the main library of Smokey's coded software that stores each of the main mathematical parts of Smokey. It includes the main functions for the stepper motors that move the parabolic dish and an infrared sensor, such as Pan_ang, Pan_ang_wh, Pan_ang_fr, Tilt_ang, Tilt_ang_wh, and Tilt_ang_fr. The "wh" stands for "whole" and the "fr" stands for fraction. Fraction is used to display the remainder of the tilt or pan angle as a character. Custom bits are stored in order for fractions to be displayed on the LCD and Bluefruit app.

```
void computeTilt(){ // calculate tilt angle from steps, make whole and fractional parts
  Tilt_ang = ((float) Tilt_steps)*0.125; // Need to measure degrees per step
  Tilt_ang_wh = int(Tilt_ang);
  Tilt_ang_fr = abs(int(10.0*(Tilt_ang - float(Tilt_ang_wh))));
}

void computePan(){ // calculate pan angle from steps, make whole and fractional parts
  Pan_ang = (((float)Pan_steps)/Steps180)*180.0;
  Pan_ang_wh = int(Pan_ang);
  Pan_ang_fr = abs(int(10.0*(Pan_ang - float(Pan_ang_wh))));

}
```

The next part in the second library includes values and functions for the LEDs. If the average value of smoke is above the warning value, the corresponding smoke LED turns on. If the average temperature is greater than the warning value and at the same time the humidity level is lower than the set value, then the corresponding humidity and temperature LED lights up. The infrared sensor's whole number displayed on the LCD screen and Bluefruit app represents the value of the strength of the infrared light, while the fraction displays the angle that the infrared light was spotted at. The third library includes all of the DO functions.

```
void computeFlm(){
  Flm_read = analogRead(Flm_pin);                  // Get current output from flame detector
//+++++ changed numbers here to make it go to zero when no flame
  Flm_det = limit( ((800 - Flm_read) / 7),0,99);// Scale it to 0 to 99
  if(Flm_det > Flm_max) {      // If its the biggest
    Flm_max = Flm_det;         //   save the value
    Flm_ang_max = Pan_ang_wh; //   and the angle.
  }
}
```

Conclusion

As a result of this work, the team has developed a prototype of a working robotic fire and fire weather conditions measuring system that has been tested in a variety of conditions and circumstances. If this prototype were mass produced and networked in the way that all of them would communicate with each other via Bluetooth, they could be then installed in the forest to serve as the wildfire protection system. Each such Smokey could use a battery that would charge during the day by a small PV solar panel and keep powering Smokey also during the night.

While working on this project the team valued the experience gained. Some team members learned how to design a circuit board and C++ programming while others learned more about video editing and presenting in front of an audience. This project helped each team member discover his/her natural interests, skills, and abilities, as well as to make their mind about the major in the college.

Collaboration

Roles / Responsibilities

Although the team shared the responsibilities and roles working on the project, their contribution depended on the certain natural skills that each team member already had. One team member was involved in research while another was in charge of tracking deadlines and putting presentations together. The team leader was mostly in charge of computer programming and working directly with the mentor, and another team member was in charge of hardware. All the team members were meeting after school to work together using their skills. The main responsibility of the team, as a whole, was to be present for meetings, presentations, and to be responsive to any communication involving the project.

Contributions

The team leader, Zachariah Burch, was regularly meeting with the mentor, and worked on designing circuit board, the C++ programming, and on the parts of the report that included the components of Smokey, code, graphs, and images.

The team leader's assistant, Isel Aragon, has been putting all the presentations together, regularly stayed in after school meetings, and worked on the report together with the team leader and their teacher.

The team member, Britny Marquez, has been participating in after school meetings and greatly contributing to the presentations of the project.

Another team member, Daniel Leon Leon, together with the team leader, was meeting with the mentor and worked on designing the circuit board.

Works Cited

"Critical mistakes and miscalculations by federal employees caused devastating New Mexico wildfire, report finds." *CBS News*, 21 June 2022, https://www.cbsnews.com/news/new-mexico-wildfire-federal-employees-critical-mistakes-miscalculations/. Accessed 10 March 2023.

Lopez, Tommy. "Las Vegas has about 30 days left of clean water." *KOB 4*, 18 August 2022, http://www.kob.com/new-mexico/las-vegas-has-about-30-days-left-of-clean-water/. Accessed 9 October 2022.

"Nmsnf Calf Canyon Information | InciWeb." *InciWeb*, 19 April 2022, https://inciweb.nwcg.gov/incident-information/nmsnf-calf-canyon. Accessed 4 April 2023.

Popovich, Nadja. "New Mexico Wildfires: Mapping an Early, Record-Breaking Season." *The New York Times*, 1 June 2022, http://www.nytimes.com/interactive/2022/06/01/climate/new-mexico-wildfires.html. Accessed 10 November 2022.

Rodriguez, Vince. "New Mexico wildfires in 2022." *KOAT*, 11 June 2022, http://www.koat.com/article/new-mexico-wildfires-calf-canyon-2022/40257066. Accessed 10 September 2022.

Romero, Simon. "The Government Set a Colossal Wildfire. What Are Victims Owed?" *The New York Times*, 24 June 2022, https://www.nytimes.com/2022/06/21/us/new-mexico-wildfire-forest-service.html. Accessed 16 November 2022.

Weeden, Meaghan. "How Trees Clean the Air." *One Tree Planted*, 5 November 2021, https://onetreeplanted.org/blogs/stories/how-trees-clean-air. Accessed 10 November2022.