

Project 2 Report

Description:

The wildfire detection and monitoring system uses Python to simulate sensor data collection for smoke and temperature levels, analyze sensor readings to detect if a wildfire is happening, log sensor activity and detection outcomes in a text file, and provide a text-based interface for accessing everything.

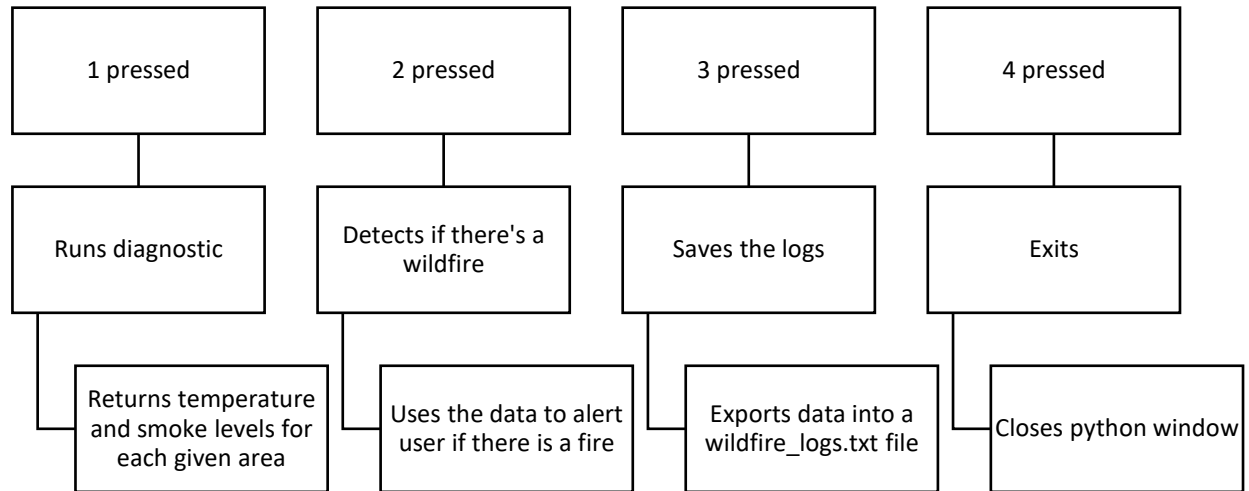
Significance of the Project:

Wildfires threaten ecosystems and cause significant environmental damage, leading to millions of dollars in damage yearly. This code simulates a simple and cost-effective detection system. In turn, making it more accessible to the general public. Because the code accounts for logging data, said data can be further studied and used to find correlations between different parameters such as wildfire risks and environmental conditions.

Installation and Usage Instructions:

Download the .py file, open the Windows terminal change the directory to where the .py file is located, then type “python wildfire_monitor.py”, then press 1 for data, 2 for wildfire detection, 3 for logging data, and 4 for exiting.

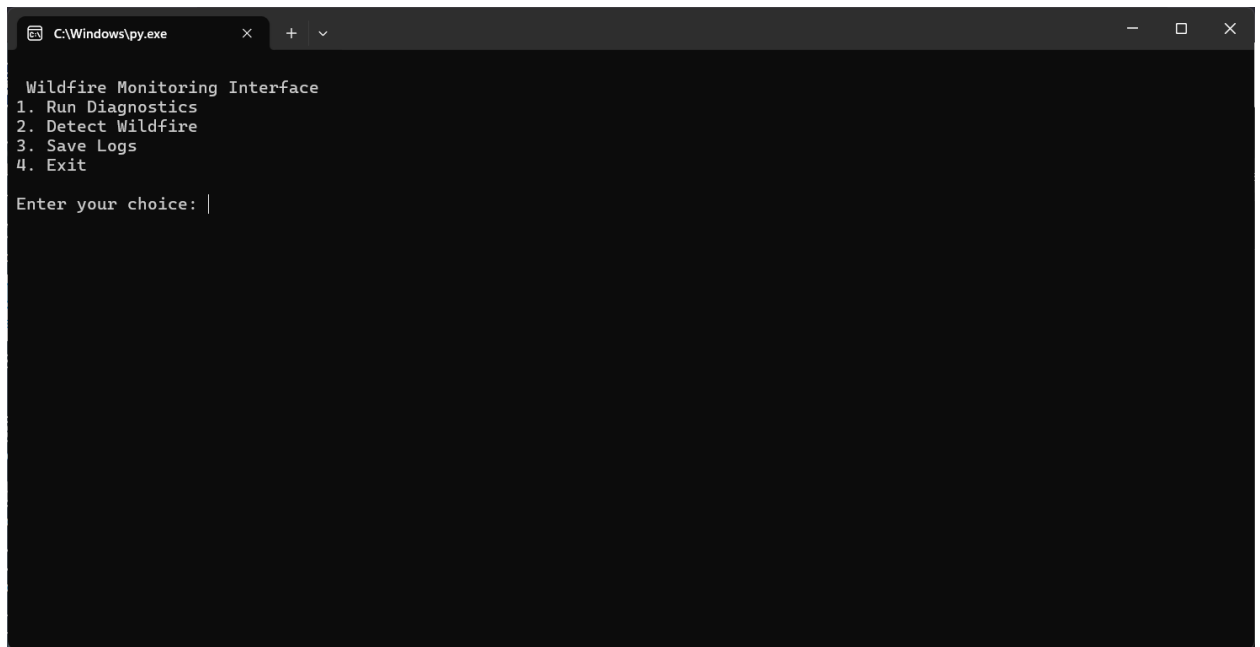
Code Structure:



Functionalities and Test Results:

The code can list the temperature and smoke levels for each area that is given, detect wildfires based on said data, and export the data to a text file for easy analysis. It can also simulate failure within a sensor and return “ERROR.”

1. Home screen:



```
C:\Windows\py.exe x + v

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: |
```

2. Running diagnostic:

```
C:\Windows\py.exe x + v

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: 1

Running diagnostics for Station PA-001...

ERROR: Temperature sensor failure at Forest A
TemperatureSensor at Hilltop reading: 42.43
TemperatureSensor at Lakeview reading: 54.33
TemperatureSensor at East Ridge reading: 59.74
SmokeSensor at Forest A reading: 3
SmokeSensor at Hilltop reading: 7
SmokeSensor at Lakeview reading: 4
SmokeSensor at East Ridge reading: 5

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: |
```

3. Detecting wildfires (if there are any):

No wildfire detected.

```
C:\Windows\py.exe x + v

SmokeSensor at Forest A reading: 3
SmokeSensor at Hilltop reading: 7
SmokeSensor at Lakeview reading: 4
SmokeSensor at East Ridge reading: 5

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: 2

Evaluating wildfire conditions...

Sensor error at Forest A: Unable to evaluate
[Hilltop] Temp: 42.43°C, Smoke: 7
[Lakeview] Temp: 54.33°C, Smoke: 4
[East Ridge] Temp: 59.74°C, Smoke: 5

No wildfire detected at this time.

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: |
```

Wildfire detected.

```
C:\Windows\py.exe
TemperatureSensor at East Ridge reading: 67.21
SmokeSensor at Forest A reading: 5
SmokeSensor at Hilltop reading: 9
SmokeSensor at Lakeview reading: 9
SmokeSensor at East Ridge reading: 0

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: 2

Evaluating wildfire conditions...

[Forest A] Temp: 94.14°C, Smoke: 5
[Hilltop] Temp: 84.68°C, Smoke: 9
🔥 Wildfire Alert at Hilltop! 🔥
[Lakeview] Temp: 55.8°C, Smoke: 9
🔥 Wildfire Alert at Lakeview! 🔥
[East Ridge] Temp: 67.21°C, Smoke: 0

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: |
```

4. Saving logs to a text file in the same directory as the .py file:

```
C:\Windows\py.exe x + v

Enter your choice: 2

Evaluating wildfire conditions...

Sensor error at Forest A: Unable to evaluate
[Hilltop] Temp: 42.43°C, Smoke: 7
[Lakeview] Temp: 54.33°C, Smoke: 4
[East Ridge] Temp: 59.74°C, Smoke: 5

No wildfire detected at this time.

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: 3
Logs saved successfully.

Wildfire Monitoring Interface
1. Run Diagnostics
2. Detect Wildfire
3. Save Logs
4. Exit

Enter your choice: |
```

```
wildfire_logs.txt x +

File Edit View

Forest A | 2025-04-28 13:03:25.847131 | ERROR
Hilltop | 2025-04-28 13:03:25.848952 | 42.43
Lakeview | 2025-04-28 13:03:25.848952 | 54.33
East Ridge | 2025-04-28 13:03:25.848952 | 59.74
Forest A | 2025-04-28 13:03:25.848952 | 3
Hilltop | 2025-04-28 13:03:25.848952 | 7
Lakeview | 2025-04-28 13:03:25.848952 | 4
East Ridge | 2025-04-28 13:03:25.848952 | 5

Ln 1, Col 1 | 354 characters | 100% | Windows (CRLF) | UTF-8
```

Showcasing the Achievement of Project Goals:

This project completes the goal because it is able to take data, process it, and alert the user of a wildfire while also allowing logging of the data. The project could be expanded upon with real temperature and smoke sensors, which would wirelessly send data to a machine with a modified version of this code to be able to read the data sent from the sensors.

Discussion and Conclusions:

The project doesn't account for temperature changes due to seasons, and it is limited by the fact that it has to use simulated data and not real-world data gathered by temperature and smoke sensors. In the project, the alert system only activates if temperature readings are greater than 50 degrees Celsius and if the smoke level is greater than or equal to 7, which could lead to false alarms or unknown wildfires happening because it doesn't account for different ecosystems. This project uses object-oriented programming (OOP), message passing with the "SensorError" class, a basic algorithm to handle the detection of wildfires, etc.