2023

Project:
Fighting fire with fire

ROBERT RAFANELLI, CLAUDIA ORTIZ, JOE BIJU, MICHEÁL GILL, RAY FOYSAL, MOAZ REFAEI

Challenge

Managing Fires: Increasing Community Based Fire Management Opportunities.

High-Level Summary

Though satellites are an important and amazing tool in our arsenal in the 21st century, having sensors on the ground is a key factor in ensuring speedy detection and addressal by emergency services. Our project intends to take on the challenge of “Managing Fire” by installing an array of sensors located through wild-fire prone areas. Each individual unit contains heat, humidity and smoke detectors that is the relayed back to a central station to be processed and displayed on a suitable graph. This real-time data can help emergency services quickly locate and stop the flames whilst it is still in its early phase. This is important in working quicker to minimise the damage to the environment and local communities. Combined with the data from satellites on terrain, weather, etc. this can be fed into a prediction algorithm which can track and forecast the spread of the fire.

Project Links

Project Demo

https://www.canva.com/design/DAFwkVmWuFw/x4a1gZBLiGtskBXygZIdHw/edit?utm_content=DAFwkVmWuFw&amp;utm_campaign=designshare&amp;utm_medium=link2&amp;utm_source=sharebutton

Final Project

https://www.canva.com/design/DAFwkVmWuFw/x4a1gZBLiGtskBXygZIdHw/edit?utm_content=DAFwkVmWuFw&amp;utm_campaign=designshare&amp;utm_medium=link2&amp;utm_source=sharebutton

Video Links

https://youtu.be/mCscuKfXkJo?feature=shared

https://youtu.be/R-zqj8HamZ0?feature=shared

Project Information Details

The prototype project consists of the following parts:

- Arduino Uno R3 + Arduino IDE
- Redbear Labs Bluetooth Shield
- DHT11 Heat and Humidity Sensor
- Carbon Monoxide Sensor
- 9 Volt Battery
- 3D Printed Housing for components using Fusion360.
- Clear Acrylic lid for Housing
- 2x Straps for securing the project to the tree.
- Oracle Database
- Android Studio
- Coding languages: Java, C++, and SQL
- NASA Global Imagery Browse Services (GIBS)

We designed this simplified prototype to show the basic requirements for such a device. The custom 3D printed housing is ergonomically designed to store all of the electrical components and also to fit any tree it could be placed on. Cavities are placed on the inside of the housing to feed the straps through to secure the beacon to the tree. On the underside of the housing, numerous slots allow heat, humidity, and carbon monoxide to enter the housing.

To know where to place these devices, we used the NASA Open Data to analyse where most forest fires have been.

This way, the sensors inside the housing read the humidity and temperature, and there is a C0 smoke sensor that detects whether if there is smoke or not. Each housing has a 'Sensor_ID' represented by:

The first letters represent the initial of the location which can be one of the following:

- Alaska (A)
- Central America and Hawaii (CAH)
- Europe, Russia, and Asia (RA)
- South America (SAM)
- Southern Africa (SAF)
- Northern and Central Africa (NCA)
- South Asia (SA)
- South East Asia (SEA)
- Australia and New Zealand (ANZ)

Following the initial of the location, there are five numbers representing a country postal code. The last 3 digits, which can range from 0 to 999, indicate the number of the device within that postal code area.

For example, if you have a sensor ID "CAH96097836," it means:

- "CAH" represents Central America and Hawaii is the location.
- "96097" is the postal code for a specific area within Central America and Hawaii.
- "836" indicates that this is device 836 created within that postal code area.

This naming convention allows for a structured way to identify and differentiate sensors based on their location, postal code, and the number of devices within that area. Moreover, as each sensor is placed manually, it saves the exact location of the housing devices.

In the future, this project could be mass-produced for a fraction of the price, economies of scale and custom-made parts will bring the price down greatly.

Doing some quick calculations, we can get an estimate for the costs of the hardware once brought in bulk.

Component	Estimated Price (€)
Temperature + Humidity Sensor	**1.20**
Custom PCB	**1.50**
9 Volt Battery	**0.30**
HDPE Injection Moulded housing	**0.25**
Straps	**0.25**
Total Cost Per Unit:	**3.50**

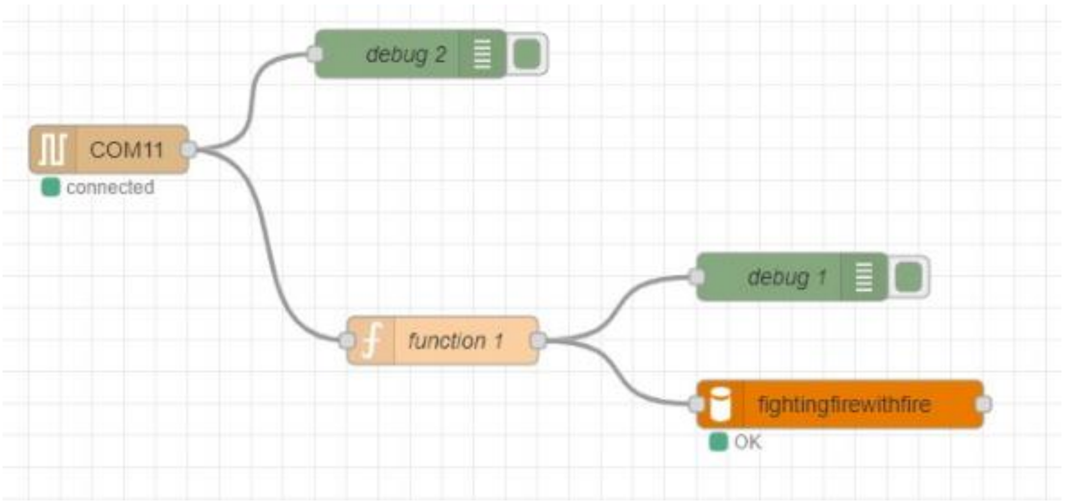
In the future, we could also send the data captured from the sensors to the database through a Wi-Fi connection (such as STARLINK), so we can send the data from thousands of kilometres away; and not Bluetooth as it is now. This way we can transfer the data without having the problem of the distance between the sensors and the database.

Connection between Arduino and database

To send the data that the Arduino reads from the sensors to the database, the process is as follows:

First, we need to run the code in the Arduino IDE that reads the data collected by the sensors. To confirm the correct reading of the data, we can open the 'Serial monitor.' However, to send the data to be stored in the database, we must close the 'Serial monitor' in the Arduino IDE and execute the 'node-red' command in the Node.js Command prompt. Next, we open XAMPP and initiate the 'Apache' and 'mySql' modules.

At this point, the Node-Red localhost and the database are both active. Within Node-Red, we need to establish the following flow connections:



By configuring these connections, we link the Serial input (Arduino) to the 'fightingfirewithfire' database. To format the collected data into a message that can be sent to the database, we use the following code:

```
// convert the json string to a javascript object
const data = JSON.parse(msg.payload);
// extract each of the readings from the payload
const {SensorID, Latitude, Longitude, Humidity, Temperature, Co2Value } = data;
// build our message
const message = {
    payload: [SensorID, Latitude, Longitude, Humidity, Temperature, Co2Value],
    topic: 'INSERT INTO sensordata(SensorID, Latitude, Longitude, Humidity,
Temperature, Co2Value) VALUES (?,?,?,?,?,?);'
}
//return
return message;
```

Afterward, we click 'Deploy' and navigate to the database's localhost page. There, we can refresh the page or click on our 'sensordata' table within the 'fightingfirewithfire' database, and we will observe the sensor data collected by the Arduino stored in the database.

Looking to the future, our next step will involve transferring the database to the cloud. This will enable us to transmit data from the Arduino to the database via WiFi, eliminating the need for a USB connection. Furthermore, by connecting multiple Arduinos to the database through a WiFi connection, we can monitor variations in temperature, humidity, and CO2 values in different areas, thanks to the coordinates of each sensor (latitude and longitude).

Space Agency Data

Global Imagery Browse Services: https://nasa-gibs.github.io/gibs-api-docs

NASA FIRMS Map: https://firms.modaps.eosdis.nasa.gov/map/#d:2023-10-07;@0.0,0.0,3.0z

NASA FIRMS: https://firms.modaps.eosdis.nasa.gov/space-apps-2023

FIRMS NASA Country Yearly Summary: https://firms.modaps.eosdis.nasa.gov/country/

NASA Earth Observatory FireMap: https://earthobservatory.nasa.gov/global-maps/MOD14A1_M_FIRE

NASA EarthData FIRMS MCD14DL-NRT: https://www.earthdata.nasa.gov/learn/find-data/near-real-time/firms/mcd14dl-nrt#ed-firms-attributes

References

Red Bear Lab Bluetooth Shield Information for Arduino: https://github.com/RedBearLab/BLEShield

Android Studio Bluetooth Permissions: https://stackoverflow.com/questions/67722950/android-12-new-bluetooth-permissions

Android Bluetooth Setup Information: https://developer.android.com/guide/topics/connectivity/bluetooth/setup

Research Connection between Arduino and database: https://icreateproject.info/2014/12/14/arduino-save-data-to-database/

Research Arduino: Display data over local network: https://icreateproject.info/2014/08/19/arduino-display-data-over-local-network/

Sending data from Arduino to PC using Bluetooth: https://forum.arduino.cc/t/sending-data-from-arduino-to-pc-using-bluetooth/372326/1

Research BLE SDK Arduino: https://github.com/Cheong2K/ble-sdk-arduino/tree/RBL

Saving Arduino Serial Data to a .txt file: https://www.youtube.com/watch?v=RWgyCcnUxPY

Research How to Connect Arduino Board to WiFi Network: https://www.youtube.com/watch?v=vRfuJ2yjEb0

Arduino SPI library information: https://docs.arduino.cc/learn/communication/spi

Research on Usual Humidity and Temperature values in Forest Fires: https://oroel.com/actualidad/la-regla-del-30-en-un-incendio/#:~:text=Qu%C3%A9%20es%20la%20regla%20del%2030%20en%20los%20incendios%20forestales&amp;text=Si%20nos%20centramos%20en%20concretar,Humedad%20relativa%20menor%20al%2030%25

Research on CO2 Air Usual values and when there is Fire: https://www.iqair.com/es/newsroom/air-pollution-and-co2-monitoring-in-schools

Map representation of data passed: https://kepler.gl/

DHT Sensor Library for Arduino: https://www.arduino.cc/reference/en/libraries/dht-sensor-library/

Adafruit Unified Sensor for Arduino: https://www.arduino.cc/reference/en/libraries/adafruit-unified-sensor/

Map data ©2023 Google: https://www.google.com/maps/

Send data from the Arduino to the SQL Database: https://www.youtube.com/watch?v=fRNKrEdmnkg