**Introduction**

Carbon monoxide is a common pollutant that is a by-product of the burning of many carbon based compounds. It is produced by many industrial processes, automobiles, and appliances. When inhaled by humans, it binds with hemoglobin in our blood to reduce the oxygen delivering capabilities of blood cells. In some countries, carbon monoxide poisoning is the most common form of fatal air poisoning. In less severe cases, heavy exposure can lead to weakness, headaches, vomiting, and more.

How can we limit our exposure to carbon monoxide? Can better data enable us to make better, healthier choices in our lives? In our project, we chose to explore the possible applications of embedded systems in environmental sensing and to gain experience building a complete and functional system that can be expanded upon in the future.

**Background**

First, to understand the problem we must present some background info on the effects of Carbon Monoxide exposure as well as levels to expect in different environments. The following data about CO exposure is pulled from Wikipedia, where it is backed by strong sources.

<table>
<thead>
<tr>
<th>Concentration</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 ppm</td>
<td>Natural atmosphere level</td>
</tr>
<tr>
<td>0.5 to 5 ppm</td>
<td>Average level in homes</td>
</tr>
<tr>
<td>5 to 15 ppm</td>
<td>Near properly adjusted gas stoves in homes</td>
</tr>
<tr>
<td>100 to 200 ppm</td>
<td>Exhaust from automobiles in the Mexico City central area</td>
</tr>
<tr>
<td>5,000 ppm</td>
<td>Exhaust from a home wood fire</td>
</tr>
<tr>
<td>7,000 ppm</td>
<td>Undiluted warm car exhaust without a catalytic converter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concentration</th>
<th>Symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>35 ppm (0.0035%)</td>
<td>Headache and dizziness within six to eight hours of constant exposure</td>
</tr>
<tr>
<td>100 ppm (0.01%)</td>
<td>Slight headache in two to three hours</td>
</tr>
<tr>
<td>200 ppm (0.02%)</td>
<td>Slight headache within two to three hours; loss of judgment</td>
</tr>
<tr>
<td>400 ppm (0.04%)</td>
<td>Frontal headache within one to two hours</td>
</tr>
<tr>
<td>800 ppm (0.08%)</td>
<td>Dizziness, nausea, and convulsions within 45 min; insensible within 2 hours</td>
</tr>
<tr>
<td>1,600 ppm (0.16%)</td>
<td>Headache, tachycardia, dizziness, and nausea within 20 min; death in less than 2 hours</td>
</tr>
<tr>
<td>3,200 ppm (0.32%)</td>
<td>Headache, dizziness and nausea in five to ten minutes. Death within 30 minutes.</td>
</tr>
<tr>
<td>6,400 ppm (0.64%)</td>
<td>Headache and dizziness in one to two minutes. Convulsions, respiratory arrest, and death in less than 20 minutes.</td>
</tr>
<tr>
<td>12,800 ppm (1.28%)</td>
<td>Unconsciousness after 2-3 breaths. Death in less than three minutes.</td>
</tr>
</tbody>
</table>
As we can see from the above tables, it is clear that carbon monoxide exposure is something to be avoided, but there are few resources for concerned citizens to discover what daily air conditions are like at a fine grained level and with data that is updated constantly.

Goals
The goal of our project is to develop a basic framework that could be used for building a large, distributed network of wireless sensor nodes that collect air quality metrics and present the information to users in a friendly and accessible manner. The project can be broken up into two parts:
1) Embedded networked sensing
2) Processing and presentation.
As realistic goals for the quarter, we decided to implement the hardware and software necessary for sensing a single gas at a time and to wireless report the data back to a server that would do very simple filtering on the data. The data is then stored into a MySQL database and presented on a web page with graphs to visualize the data. We originally would have liked to have developed an Android based client for accessing this data, but cut this feature to meet our set deadlines. We hope to come back to this and develop a useful application for users to query live data from their phone in a simple manner.

Development
The fun of the project is that it involves all sorts of development: wiring sensors, Java on a mobile device, a back end server for processing, database programming, as well as web development. This provided us with an enjoyable challenge of being able to build a complete system with many components. With so many different tools, languages, and environments, it was difficult at times having to learn new tools and tricks, but this also proved a rewarding experience.

Our embedded device was a Sun Microsystems SunSPOT. It has a 32-bit ARM920T core running at 180MHz, 512K RAM, 4M Flash, and a 2.4GHz 802.15.4 radio. It runs a Java VM and allows Java applications to be deployed to the device. It includes a simple sensor board with some LEDs, accelerometer, and pins for connecting an analog signal to the on board ADC. We purchased simple and cheap ($5) Carbon Monoxide (Fig. 1) and Methane gas sensors (Fig. 2) from Sparkfun Electronics, which after a few days of reading and soldering, we had ready for wiring to the SunSPOT. We then developed a SunSPOT application that handled the sensed data as well as establishing a connection with nearby nodes to broadcast its sensed data to propagate through the network to the base station. The base station is a SunSPOT (without a battery) that is connected via USB to a laptop machine. Once the data reaches the base station machine, filtering is performed on the data to remove any obviously erroneous data points before storing it. This filtering is relatively easy to do: gas sensed data changes slowly over time and we should never see any sudden jumps or drops in measurements unless the device has become damaged. Therefore we only keep values that are within some adjustable range of the previously recorded data for that sensor. If we determine that the point is valid and worth keeping, we store it into a MySQL database, which in our experimental setup was located on the same base station machine. We keep a table of all approved sensors in the system, including their longitude and latitude position, as well as a table of data which stores the node id, a

![Image of sensors](image-url)

1. Carbon Monoxide sensor  2. Methane sensor
3. Breakout board
timestamp for the reading, and a value. This same base station machine then runs an Apache web server which polls data from our database and presents a map of sensor locations, as well as a graph of recently sensed data. We got in contact with Sun regarding a service they are testing called Sensor.Network for the sharing and visualization of data and started using that service for our data storage. Their service is far from complete but gave us some ideas for how to best present data and to create a platform that allows sensors to be added and contribute data easily.

**Results**

The ADC readings on the SunSPOT are the $V_{load}$ readings. To calculate the PPM (CO) we need to use the following equations which can be derived from the Kirchoff's current law and from the graph 1 respectively. The graph 2 shows the estimated PPM (CO) plotted with Time.

1. \[ \frac{R_s}{R_{load}} = \frac{(V_c - V_{load})}{V_{load}} \]  
   \[ \text{(R}_{load}\text{)=10KOhm, } V_c=5V \]

2. \[ 0.9(PPM) + 3900\left(\frac{R_s}{R_o}\right) = 3990 \]

![Graph 1. Rs Vs PPM (CO)](image)

![Graph 2. PPM (CO) Vs Time](image)

**Problems/Lessons**

During the course of the project, we faced the following mentioned problems.

**Calibration:** To calibrate the sensors, we had to take the sensor readings by placing the sensor at a constant 100ppm CO concentration. This is possible only if we had a constant source of CO in a closed environment which is hard to find. After some research, we found that the Air Pollution Control District County of San Diego maintains hourly reading of pollutants like CO from various places on their website. So we planned to coordinate with them to calibrate our sensors. But then we were surprised to see that their CO readings were less than or around 1ppm (apparently, San Diego is a very clean city). The problem is that our sensor can detect CO levels only above 20ppm. So, we end up realizing that we can’t use these readings for calibration. Instead, we are now planning to buy or lend a CO detector and use its readings in a highly polluted CO environment to calibrate the sensor.

**Temperature:** The sensor reading is dependent on the temperature and we have to figure a way to negate this. After going through the documentation of the sensor we found that the manufacturer has provided a graph of Temperature VS Output voltage for the sensor. But, the graph is a curve whose equation is unknown and hence we need to use mathematic techniques to find the best approximate equation for the curve using which we can negate the change in readings due to temperature.

**Sensor technology:**

The different sensor technologies available in the market are

1. Semiconductor sensors
2. Electrochemical sensor
3. Infrared sensor
Different technologies have different detection levels, reliability and maintenance requirements. The semiconductor sensors are small, easy to use, reliable and have longer life time but they don’t detect the whole span of concentration levels. Our sensor is a semiconductor sensor which explains why it can only detect if the concentration levels are above 20ppm.

The Electrochemical and Infrared technologies sensors are more accurate and span a wide range of concentration levels but the problem is they have a short life time and are much bigger in size compared to the semiconductor sensors and hence making them unsuitable for most of the embedded applications.

So, there is a tradeoff between size and span of detection. Is there a way in which I could utilize the advantages of both these technologies? It is with this goal in mind that we are thinking of building a hybrid system which can use both the type of sensors.

Radio:

The radio in the SunSPOT is not very powerful. It has a range of 10-15m at maximum power inside the buildings. So, while building a custom board in the future, we have to keep this in mind before choosing a radio.

FUTURE PLANS

The short term plans of the project include calibrating the sensor correctly, finding the temperature- voltage dependence equation.

The long term plans include developing a hybrid sensor network system which would effectively utilize the advantages of both the types of sensors technologies without affecting the practical applicability.

Further we would use this gathered data to provide it to the users on cell phones like Android etc. Later this would evolve to provide automated suggestions about the route to be followed to avoid pollution.

Resources

SunSPOTWorld – http://www.sunspotworld.com/

Community for SunSPOT development

SparkFun Electronics – http://www.sparkfun.com/

Designs and provides boards and sensors for projects of all sizes

Components used:


Resource for storing, sharing, and visualizing sensor data

Limited beta. Used for our graphs.

United States Environmental Protection Agency - http://www.epa.gov/

Provided statistics and information about Carbon Monoxide