ECE 445
Spring 2016

Sound controlled smoke detector

Project Proposal

Yihao Zhang, Xinrui Zhu, Meng Gao

2016-Feb-8

Index

Introduction

Statement of Purpose

Almost all of us encounters false fire alarms at some point in our life. Although many modern fire alarms have the ability to mute temporarily with a push of button, the physical location of the smoke detector does not always make it easy to do so. At the same time, voice controlled products are entering markets and gaining popularity in recent days. These products, such as Android phones and Amazon Echo, can be activated by keyword such as “OK Google”, ”Alexa” or “Amazon”.

Therefore, we propose a sound controlled fire alarm that allows users to easily turn the alarm off by shouting the keyword "cooking" when false alarm happens (in addition to a push button). In specific, we plan to use many different human version of “cooking” as the training data. Then the processor will find and store the Mel-Frequency Cepstral Coefficients (MFCCs) for the training word. Then once the alarm is triggered, the core will be turned on and actively listens for the keyword, finding the MFCCs for what it hears, and comparing with the stored MFCCs. If the mean square error is below a threshold, the core will stop the alarm. In additional, we may also look into Dynamic Time Warping (DTW) to improve our detection accuracy.

Objectives

Goals and benefits:

- Allow false fire alarms to be safely turned off by shouting the keyword "cooking"
- Prevent unnecessary interruption to everyday activities such as cooking
- Maintain sufficient warning against possible fire hazard.

Functions and features:

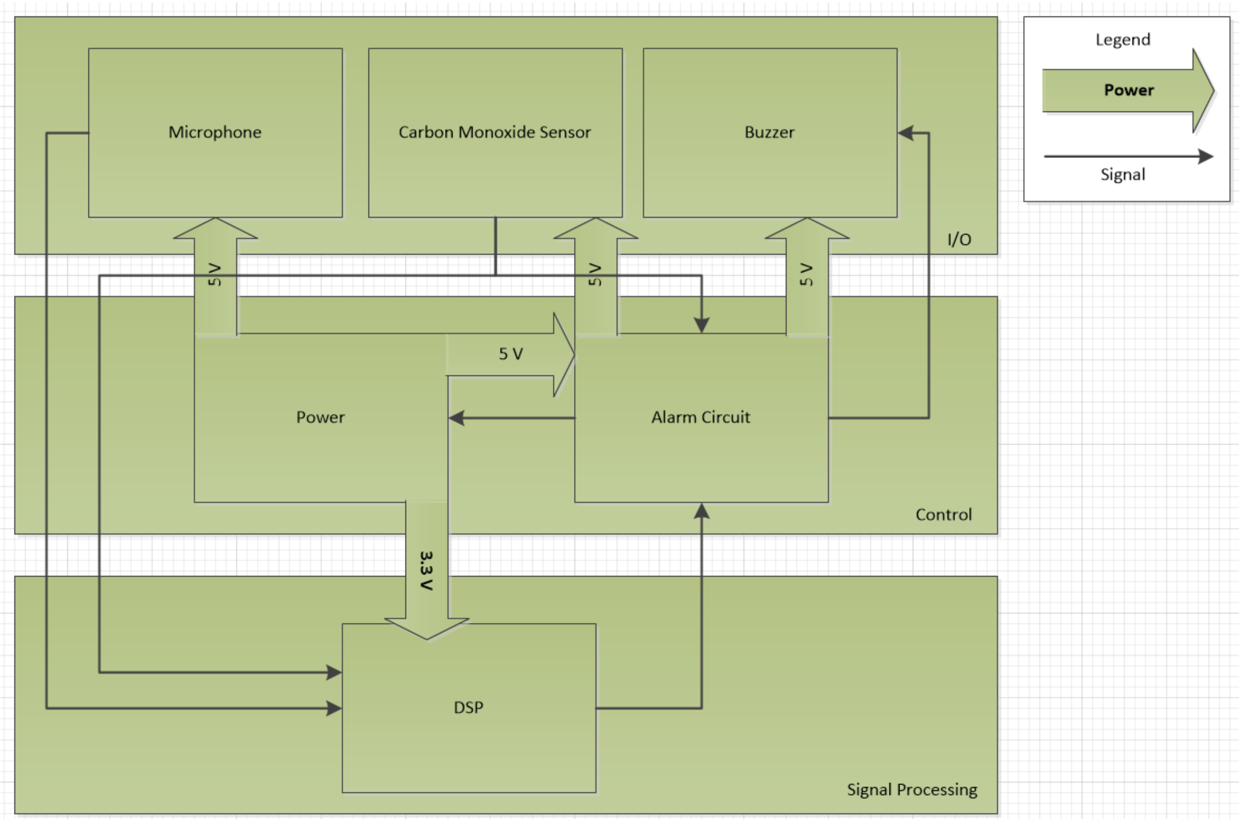
- Sense environment condition relating to fire, specifically, Carbon Monoxide concentration.
- Activate alarm and power on Digital Signal Processor (DSP) if fire hazard is detected.
- Capture human voice if alarm is triggered
- Implement an algorithm to recognize keyword by feature extraction using Mel-Frequency Cepstral Coefficients
- Process audio data in real time using our algorithm and check if there is a match to the keyword
- Interpret CO sensor data to calculate ppm
- Pause the alarm if keyword is matched and if it is safe to do so

Design

Block Diagram

Figure 1: Block Diagram

Block Descriptions



Power

This block will supply 3.3V to power the DSP, and 5V to power the circuit and sensors.

Microphone

We will use that microphone for Arduino that is available off-the-shelf. The microphone will capture voice, and send analog signal to the DSP.

Carbon Monoxide Sensor

We will use MQ-7 sensor to detect carbon monoxide density in air, this sensor will be obtained off-the-shelf. If the CO concentration is above threshold (around 70 ppm), the alarm will be triggered by the alarm circuit. The analog output from the sensor will also be fed into the DSP, and allows the DSP to consider the current CO concentration before suppressing any alarm.

Buzzer

This is the alarm that will sound if fire is detected. The buzzer will be obtained off-the-shelf. It is controlled by the alarm circuit.

Alarm Circuit

The alarm circuit will be built around a timer IC that can be configured to oscillate at 3 kHz. If the sensor detects CO concentration above active threshold, this circuit will trigger the alarm at 3 kHz. The circuit should also be capable of pausing the alarm when told to do so by outside signal.

Test switch and pause switch should be provided in this circuit to allow testing both in the design process and by user without actually starting a fire, where test switch simulates the sensor input, and pause switch simulates the DSP input.

DSP

We will use TI MSP430 as our digital signal processor. The DSP will listen for voices through the microphone, and runs MFCCs to identify the keyword "cooking". Once the keyword is matched, the DSP calculates the current CO concentration from the sensor, and turns off the alarm if the current CO level is below 100 ppm.

Requirements and Verification

Table of Requirements and Verification:

Name	Requirements	Verification
Microphone Buzzer CO Sensor	Functions for 4.5 to 5.5V	Off-the-Shelf components Manufacturer verified
Power	a. Output at 3.3V ± 0.3 V for DSP b. Output at 5V ± 0.5 V for alarm circuit and microphone	1. Attach oscilloscope across power source 3.3V terminal 2. Ensure output voltage remains between 3V to 3.6V 3. Attach oscilloscope across power source 3.3V terminal 4. Ensure output voltage remains between 4.5V to 5.5V
Alarm Circuit	a. Able to trigger alarm b. Able to pause alarm c. Trigger alarm when there's fire	A) 1. Make sure power is on 2. Attach oscilloscope at the timer chip output 3. Press test button on alarm circuit to force an alarm 4. Timer should oscillate at 3 kHz B) 1. Make sure power is on 2. Attach oscilloscope at the timer chip output 3. Press test button on alarm circuit to force an alarm 4. Timer should oscillate at 3 kHz 5. Use pause button on alarm circuit to pause alarm 6. Timer should stop oscillation C) 1. Go outside 2. Make sure power is on 3. Light up a cigarette 4. Bring cigarette close to smoke sensor 5. The alarm should start
Voice Recognition Algorithm	a. Able to recognize keywords with no false positives, and achieve above 60% accuracy.	1. On a PC, open Matlab 2. Copy/Paste test files and scripts to working directory 3. Run mfcc_test() 4. Verify result as printed on terminal
DSP	a. Able to send an active high signal to pause alarm by	A) 1. Make sure power is on for DSP

	responding to dsp test button b. Able to run the voice recognition algorithm and detect keyword	2. Attach an LED to DSP output GPIO 3. Apply +5V at DSP test input GPIO 4. LED on the output should light up B) 1. Make sure part verification A) is working 2. Make sure power is on for both mic and DSP 3. Attach an LED to DSP output GPIO 4. Shout "cooking" at 5 meters away 5. LED should light up, if not, try repeat step 3 a few more times.

Tolerance Analysis:

The most important block in our design is the implementation of voice recognition algorithm. This is because this algorithm ultimately decides if our fire alarm will disregard a potentially life threatening condition and stop warning the user about it. Therefore, we have absolutely zero tolerance for false positives, and we never want to falsely turn off a fire alarm. On the other hand, while we strive to catch every keyword, we are certainly able to tolerance a lot more false negative, where the user may just need to shout the keyword one or two more times when fire alarm triggers. In fact, even with 60% recognition, there can be as much as 0.6+0.4*0.6+0.4*0.4*0.6=93.6% possibility that the keyword will be detected in less than three tries.

Being aware of the difference in tolerance for false negative and false positive, we can make trade-off by lowering our matching threshold, thus lower overall accuracy but eliminates false positives as much as possible. Moreover, we send the CO sensor output to the DSP as well. This allows the DSP to check and calculate the precise concentration of CO in air before it turns off the alarm. This not ensures fail safe operation in case of false positive, it also protects the user from falsely turning off the alarm or anyone from intentionally tampering with the alarm.

Cost and Schedule

Labor

Name	Hourly Rate	Hours	Total = Hourly X Hours X 2.5
Meng Gao	35	165	14437.5
Yihao Zhang	35	165	14437.5
Xinrui Zhu	35	165	14437.5
		495	43312.5

Parts

Parts	Quantity	Total
Carbon Monoxide Sensor MQ-7	1	5
Buzzer CEM-1203(42)	1	4
Microphone CEM-C9745JAD462P2.54R	1	10
Digital Signal Processor TI MSP 430	1	10
Timer IC TI LM555	1	1
Resistors, Capacitors, Inductors	N/A	10
Total		40

Grand Total

Section	Total
Labor	43312.5
Parts	40
Total	43352.5

Schedule Sheet

Week of, (important dates)	Tasks
2/14 (Eagle assignment due 2/19)	Prepare for Mock Design Review - Everyone
2/21 (Design Review Signup 2/22) (Lab safety training due 2/24) (Soldering assignment due 2/26)	Prepare for Design Review - Meng Gao Start alarm circuit design - Meng Gao Obtain hardware - Meng Gao Research MFCC, DTW, and Learn Matlab - Yihao Zhang, Xinrui Zhu
2/28 (Design Review)	Matlab Model - MFCC and voice recognition - Xinrui Zhu Matlab Model - DTW - Yihao Zhang Build and test alarm circuit - Meng Gao
3/6	Get familiar with DSP - Everyone

	Matlab Model - test, debug, and precision analysis - Xinrui Zhu Microphone communication with DSP - Meng Gao, Yihao Zhang
3/13	Implement MFCC on DSP 1) Pre–emphasis -Yihao Zhang 2) Framing - Yihao Zhang 3) Hamming windowing - Yihao Zhang 4) Fourier transform - Xinrui Zhou Power System - Meng Gao
3/20 (Spring Break) (PCB first revision 3/23)	Spring Break: No additional work allocated Wrap up any unfinished work Get head start on later work
3/27 (R&V 2nd Attempt due 3/28) (Individual progress reports due)	Implement MFCC on DSP 5) Mel Filter Bank Processing - Xinrui Zhou 6) Discrete Cosine Transform - Yihao Zhang Hardware integration, test, and debug - Meng Gao
4/3 (R&V Final Attempt due 4/8)	Integrate, test, debug voice recognition on DSP - Xinrui Zhu Circuit connection functionality checking - Yihao Zhang Alarm system integration, test and debug - Meng Gao
4/10 (Mock Demo during TA meeting) (Revised PCB 4/11)	According to the respond of TA to fix some problem in the project and update it. - responsibility TDB
4/17	Retest & prepare for demo - Yihao Zhang Start on final paper(software & algorithm) - Xinrui Zhu Start on final paper(hardware & circuit) - Meng Gao
4/24 (Demo)	Finish Final Paper(software & algorithm) - Xinrui Zhu Finish Final Paper(hardware & circuit) - Meng Gao Finish Final Paper(everything else) and prepare for presentation(PPT) - Yihao Zhang