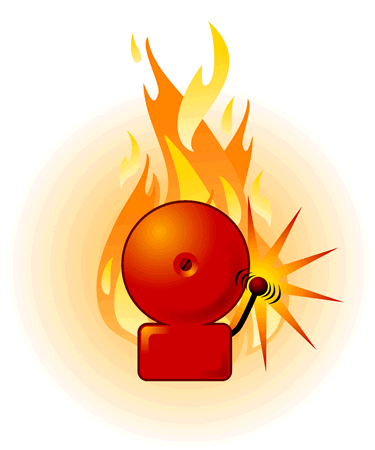
**FIRE DETECTION SYSTEM**

-Stuti Ajmani 2008051

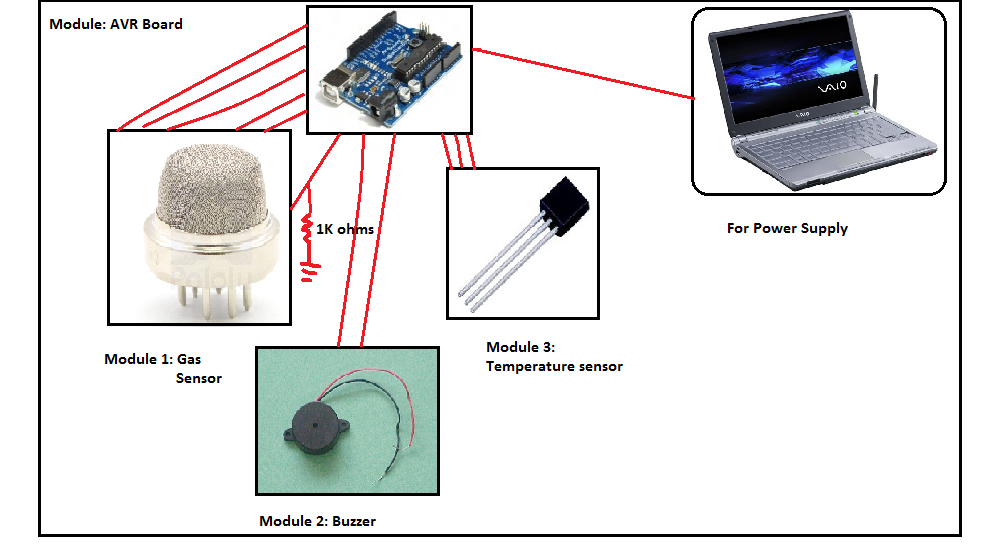


**SYNOPSIS**

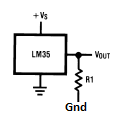
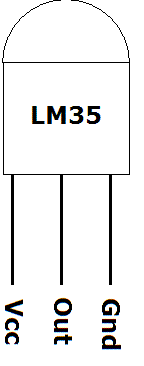
This is a fire detection system that uses temperature sensor and smoke sensor to detect fire. In case of fire, the temperature of the surroundings changes abruptly. The temperature sensor can observe this change and signal. Smoke sensors detect the concentrations of combustible gas in the air. The sensor can easily measure concentrations of flammable gas in the range of say 300 to 10,000 ppm. When these sensors are interfaced with the AVR board (with Atmega328p microcontroller), the ADC(Analog to digital convertor) converts these analog signals to digital signals. In order to increase the sensitivity of the fire detection system, if even one sensor signals firean alarm rings and an LED lights up.

The fire detection system has been tested with cables of different lengths. This is a substantial part as the resistance of the cable wire is expected to change with the length of the wire. Longer the wire, more is the resistance.

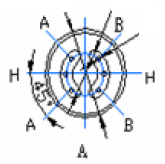
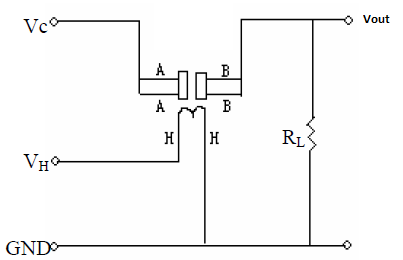
**ARCHITECTURE (Modular Level)**



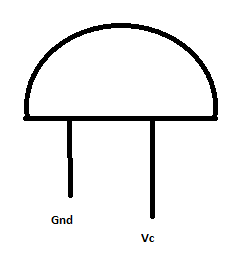
**HARDWARE INVOLVED**



LM35 has 3 pins. One is connected to ground or a pin with output as 0V on the AVR board. Second pin is connected to 5V and the third pin is connected to the ADC, at PortC pin 2, i.e. works with ADMUX =4.



The MQ2 gas sensor has 6 pins. The two pins marked A are connected to pins with output as 5V, ie. Vc. There are 2 H pins, 1 is connected to the ground and the other is given a voltage of 5V. The remaining 2 pins which are marked B, contain the output. They are both connected to the ADC through pin 0 of PORTC. A load resistance of 1K ohms is connected such that it one of its legs has the same voltage as the output voltage and the other pin is connected with ground.



The buzzer has 2 legs. One is connected to ground and the other to a output pin on the AVR board, such that the buzzer makes a sound of constant frequency till the time that pin is high.

|  |  |  |
| --- | --- | --- |
| **HARDWARE ITEM** | **COST (In Rs.)** | **PROCURED FROM** |
| Smoke Sensor MQ2 | 450 | Rhydolabz.com |
| Temperature sensor lm35 | 55 | Rhydolabz.com |
| Buzzer | 20 | Lajpat Rai Market |
| Connecting wire (8m) | 40 | Lajpat Rai Market |

**SOFTWARE**

The software component of the project is not very complicated.

It begins with the initialization of all the required registers. Thereafter I have the code for the sensors to detect smoke and change in temperature. The difference between any 2 successive readings is calculated to detect fire. If the difference is larger than the threshold specified in the code, the buzzer sirens to claim fire. The buzzer is made to beep by making the pin 0 of port B high. This threshold was set to 20 for the smoke sensor after taking readings multiple times, for approximately 3 mins. and then calculating the standard deviation. This threshold was fairly constant for wires of all lengths as the sensor shows a sudden change in value when exposed to smoke. On the other hand, without smoke the value read by the sensor keeps changing (though very very gradual) and the range of values is higher for values of greater lengths. For the temperature sensor, if it reads a temperature higher than 50 degree Celsius the buzzer buzzes.

I tested the sensors with wires till length 6m. The performance did not vary very prominently but very slight changes were there.

**LEARNINGS**

1) How to deal with problems like:  
 a) mode is not an external or internal batch command.  
 b) AVR not in sync.  
 c) STK500 no response. Expected STK—

2) The importance of load resistance. During the assignments we could do without the load resistance even if it was required according to the datasheet. I learnt that devices like load resistance, capacitors etc. if mentioned in the circuit in the datasheet should not be avoided. They change/improve the output by a great deal. Not using them can even damage the sensor you are trying to interface.

3) The essence of DC characteristics provided in the datasheet.

4) Ways to work even when there is lack of hardware resources. Example:

a) There were no male connectors and I needed extra 5V and ground pins on the AVR board.  
 b) Soldering without the soldering machine!!

5) While the code is being burnt, the RX and TX pins should be free.

6) Two numbers of different data types, i.e. unsigned int and int should not be subtracted.

7) A delay between two conversions is required, else random readings ensue.

**CHALLENGES**

**1)** The gas sensor has very thick pins. As a result I could not connect them to the pins on the AVR board. I could not arrange thick crimps or the breakout board of the sensor. So I faced a very big challenge interfacing this sensor. I was finally successful after trying all sorts of techniques like soldering bare wires to the pins of the gas sensor, thickening the holes of the bread board, soldering crimps on the pins, using tapes and so on but none of these worked!!

**2)** Using wires of different lengths. It required changing too many connections very often. Moreover since the pins of my gas sensor were not properly connected with the AVR board using crimps I had a lot of problem even disturbing the gas sensor a little. This turned out to be even more difficult when I had to keep changing wires to try with wires of different lengths.

**3)** The testing phase was very challenging as I had to deal with fire and smoke. It became even more challenging when my sensors were not placed close to each other and I wanted both (i.e. smoke and temperature sensor) of them to detect fire.

**REFERENCES**

1) MQ2 datasheet: <http://www.pololu.com/file/download/MQ2.pdf?file_id=0J309>  
 <http://www.seeedstudio.com/depot/datasheet/MQ-2.pdf>

2) LM35 datasheet: <http://www.national.com/ds/LM/LM35.pdf>