VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
(VJTI)
MATUNGA, MUMBAI 400 019

Automatic Smoke and Gas Detector using 8051 Microcontroller

Date – 6th April 2019

Abstract

The objective of this project is to control the gas supply in a given industrial of commercial environment depending on whether there is excess presence of gas (in ppm) in the given environment suggesting the possibility of a leak. Majority of fire incidents caused in industrial and commercial settings are caused due to electrical fires started due to presence of any inflammable substance majority of them being various combustion gases used as fuels in these settings. The solution suggested employs use of a gas sensor to detect the levels of gas in the environment and monitor them and if the levels exceed the prescribed limits then to signal a given microcontroller circuits to enact protective countermeasures like indicating presence of gas, sounding the alarm and closing the gas supply to prevent excess gas leakage.

Introduction

Combustion gases are instrumental as fuels in modern industrial and commercial sectors where they are the majority substances used as fuels. Also electricity is a major resource in these areas helping to drive and power majority of the processes that are undertaken. But majority of industrial accidents are also caused because of mishandling of these fuel gases and electricity. One of the leading causes of fire, be it in a restaurant kitchen or an iron smelting plant is spontaneous ignition of leaked inflammable gases when they come in contact with any sort or spark or flame. One way to solve this problem is to have an early detection and warning system that detects the presence of excess quantities of certain gases in the workspace and alerts workers of such conditions and also takes protective countermeasures such as switching off the gas and electrical power supply, which can be controlled by a central main switch or a valve, in many facilities.

Concept and Working

To realize a real-time, real-world design to sense dangerously high levels of gas and take protective steps to prevent any mishaps we use an electronic circuit consisting of an 8051-family microprocessor (AT89S52), which is interfaced with a LCD display to show status of the sensing unit as well as the control valve, and also with a Gas Sensor (MQ-2) via an 8 pin analogue to digital convertor (ADC0804) to sense the valve of gas in the workspace environment and give a reading in Parts Per Million (ppm).

Various gas sensors are available in the market which sense various different kinds of gases like butane, methane, carbon monoxide, LPG, CNG etc. In this project we are going to use the MQ-2 gas sensor as it can detect various gases like butane and other LPG related gases as well as smoke. Hence the device will also act and a gas sensor as well as a smoke/fire detector.

The gas sensor senses the presence of and gas/smoke particles in the air and give an output in digital format as high is gas is present and low if gas/smoke is absent. The gas sensor is also capable of giving precise output of the amount of gas particles present in air in analogue format.

To get the precise reading in form of a particular ppm value we have to connect the gas sensor to an analogue to digital convertor, which converts the analogue value of a differentiating voltage received from the gas sensor into a Digital value in binary format. As we are using an 8 bit ADC in this project the readings from the gas sensor will vary between 256 different values. (as 2^8=256)

The reference voltage of the ADC is adjusted so that the values of ppm to be shown vary from 0ppm to 999ppm.

For a specific value of gas in ppm (just below the prescribed dangerous levels), the microcontroller should trigger the alarm buzzer and send command to shut the gas supply valve. Also appropriate message should be displayed on the LCD screen.

After the alarm has been triggered when it has detected smoke or gas and the supply valve has been shut down, if the value of the stimulant decreases below the prescribed limit, the control valve still stays shut to prevent further leakages by opening valve again, this system should only be reset after it has been made sure that the required repairs have been done and it is perfectly safe for normal operation to consume again.

Components

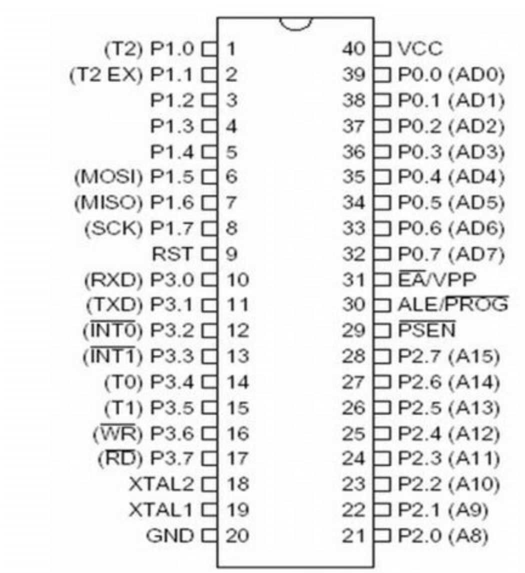
Microcontroller –

Table 1: ATMEL 89S52 microcontroller

Flash Memory Size	8192 B
Clock speed	24.0 MHz, 33.0 MHz
Supply Voltage (DC)	5.00 V, 5.50 V
Lead-Free Status	Lead Free
Packaging	Tube
Case/Package/Footprint	DIP-40
Memory Size	8000 B
RAM Memory Size	256 B
Access Time	240.0μs
Mounting Type	Through Hole
Number Of I/O Pins	32
RoHS	Complaint
No. of Pins	40

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8Kbytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry-standard 80C51 instruction set and pin out. The on-chip flash allows the program memory to be reprogrammed in-system or by a conservative non-volatile memory programmer. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry.

AT89S52 Microcontroller-



<u>ADC</u>-

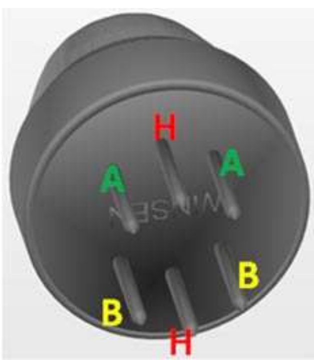
ADC0804 is a 20-pin Single channel 8-bit ADC module. Meaning it can measure one ADC value from 0V to 5V and the precision when voltage reference (V_{ref} –pin 9) is +5V is 19.53mV (Step size). That is for every increase of 19.53mV on input side there will be an increase of 1 bit at the output side.

Since the IC comes with an internal clock we do not need many components to make it work. However to make the internal clock to work we have to use a RC circuit. The IC should be powered by +5V and the both ground pins should be tied to circuit ground. To design the RC circuit simply use a resistor of value 10k and capacitor of 100pf (approx) and connect them to CLK R and CLK IN pins as shown in the circuit below. The chip select (CS) and Read (R) pin should also be grounded. The V_{ref} pin is left free because by default without any connection it will be connected to +5V.

The digital output will be obtained from the pins DB0 to DB7 and the analog voltage should be connected to V_{in}(+) pin as shown in the circuit. Also note that another end of the voltage source (sensor/module) should also be grounded to the circuit for the ADC conversion to work. Now, for the ADC Conversion to start we have to make the Write(WR) pin to go high momentary this can be done connecting the pin to I/O of MPU and toggling it high before every ADC read. Only if this is done the ADC value on the output side will be update

Gas Sensor (MQ-2) –

Resistance of sensor(Rs): Rs=(Vc/VRL-1)×RL



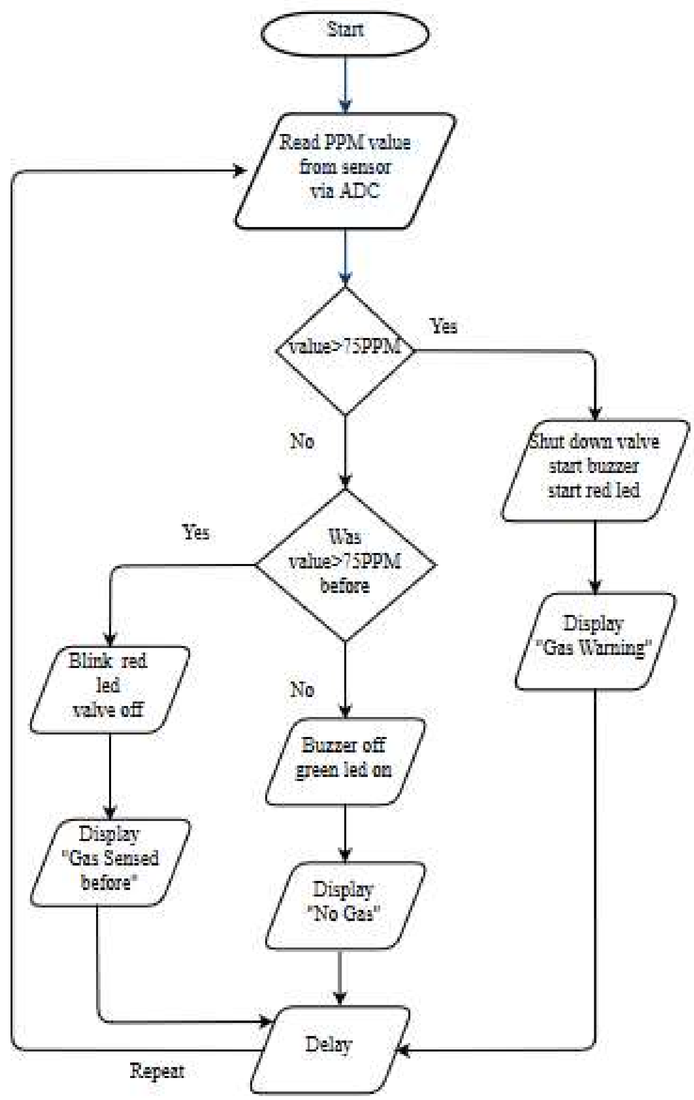
Pin No.	Pin Name
1	Vcc(+5V)
2	Ground
3	Digital Out
4	Analog out

MQ-2 Gas Sensor (Front, Back)

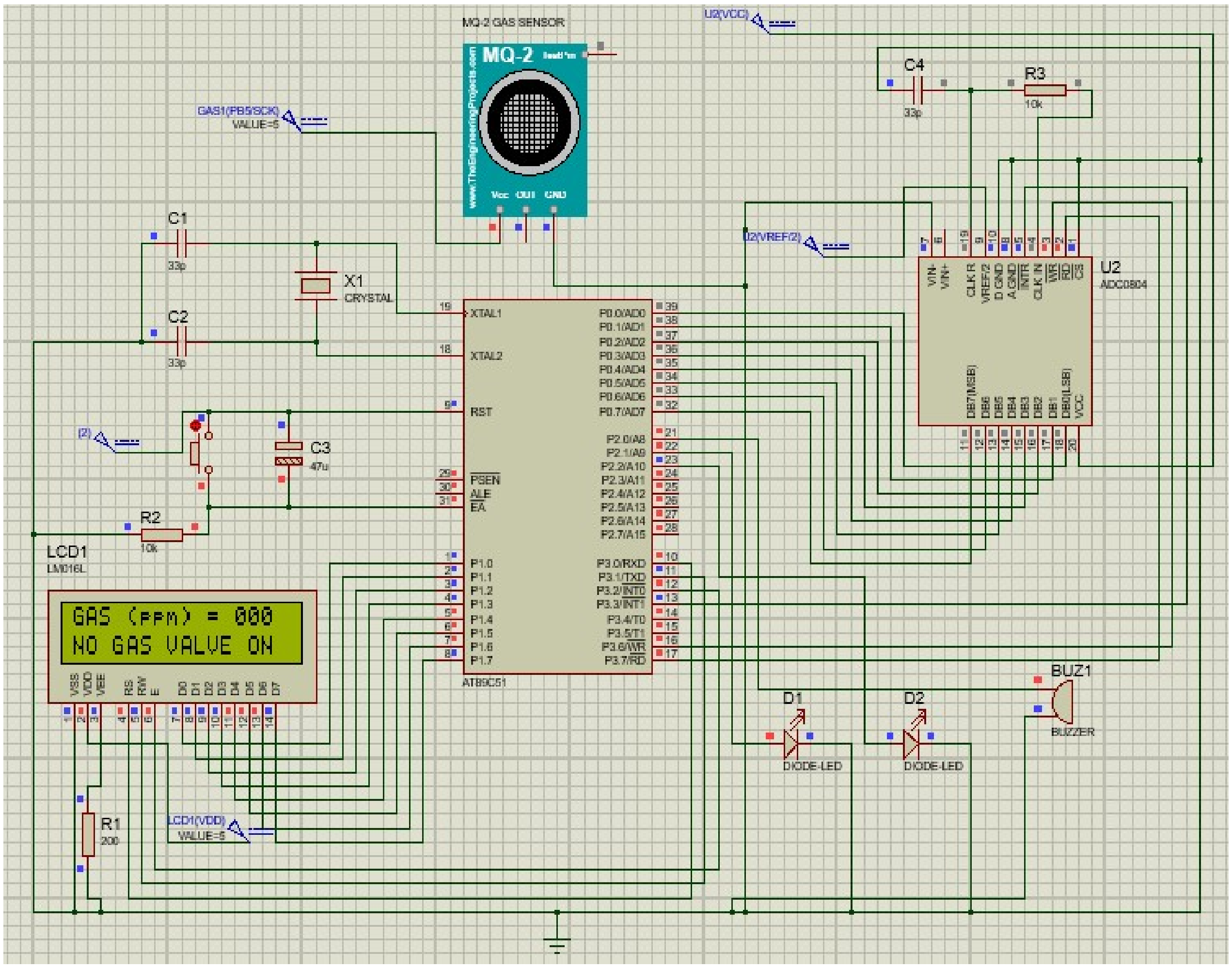
The MQ-2 Gas sensor can detect or measure gasses like LPG, Alcohol, Propane, Hydrogen, CO and even methane. The module version of this sensor comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when trying to detect one particular gas. When it comes to measuring the gas in ppm the analog pin has to be used, the analog pin also TTL driven and works on 5V and hence can be used with most common microcontrollers.

The operating voltage of this sensor is +5V. It can be used to measure or detect LPG, Alcohol, Propane, Hydrogen, CO and even methane. It's analog output voltage ranges from 0V to 5V whiles it's digital output voltage varies from 0V or 5V (TTL Logic). This sensor requires preheating before normal operation and has a preheat duration 20 seconds. This can be used as a digital or analog sensor. The sensitivity of digital pin can be varied using the potentiometer.

Flowchart



Circuit Diagram



Program Code

```
#include <REGX51.h>
sbit rs=P3^0;
sbit rw=P3^1;
sbit en=P3^2;
sbit rd=P3^7;
sbit wr=P3^6;
sbit intr=P3^3;
sbit buzzer=P2^0;
sbit ledgreen=P2^1;
sbit ledred=P2^2;
//sbit cs=P2^6;
void lcdstart();
void commandLine(unsigned int a);
void dataLine(unsigned int b);
void display(unsigned int c);
void delay(unsigned int c);
unsigned char adc(),reading,value;
int gasflag;

// Following function initializes the command line cursor for the display
void commandLine(unsigned int a)
{
	P1=a;
	rs=0;
	rw=0;
	en=1;
	delay(1);
	en=0;
}

// Following function initializes the LCD display
void lcdstart()
{
	commandLine(0x38);
	commandLine(0x0e);
	commandLine(0x06);
	commandLine(0x01);
	commandLine(0x80);
}

// Following function initializes data buffer for the LCD display
void dataLine(unsigned int b)
{
	P1=b;
	rs=1;
	rw=0;
```

```
        en=1;
        delay(1);
        en=0;
}

// Following function creates the data to be printed on the display
void display(unsigned char *p)
{
        while(*p)
                dataLine(*p++);
}

// Following function enables voltage conversion in ADC
unsigned char adc()
{
        //cs=0;
        wr=0;
        rd=1;
        wr=1;
        while(intr==1);
        //cs=0;
        rd=0;
        value=P0;
        rd=1;
        //cs=1;
        return value;
}

// Following function controls the delay between each refresh of the LCD display in ms
void delay(unsigned int c)
{
        int i;
        int j;
        for(i=0;i<c;i++)
        {
                for(j=0;j<1250;j++);
        }
}

void main()
{
        lcdstart();
        P0=0xFF;
        //P2=0x00;
        while(1)
        {
                display("GAS (ppm) = ");
                reading=adc();
                while(intr==1);
```

```
			commandLine(0x8c);
			dataLine((reading/100)+48);
			dataLine(((reading/10)%10)+48);
			dataLine((reading%10)+48);
			commandLine(0xc0);
			// If value of Gas to be sensed is above safe levels
			if(reading>75)
			{
				display("GAS, VALVE OFF");

				buzzer=1;
				ledred=1;
				ledgreen=0;
				delay(100);
				lcdstart();
				buzzer=0;
				gasflag=1;
				//break;
			}
			else
			{
				// If value of Gas to be sensed has never breached safe levels during current operation
				if(gasflag==0)
				{
					display("NO GAS VALVE ON ");
					delay(1);
					ledred=0;
					ledgreen=1;
				}
				// If value of Gas to be sensed has breached safe levels during current operation but is
				currently below safe level
				else
				{
					display("VALVE OFF");
					delay(100);
					commandLine(0x80);
					display("GAS SENSED BEFOR");
					commandLine(0xc0);
					display("INITIATE REPAIRS");
					buzzer=1;
					ledred=1;
					ledgreen=0;
					delay(100);
					lcdstart();
					ledred=0;
				}
			}
		}
	}
```

Safety Concern

As this design doesn't switch itself off after sensing inflammable gas or smoke to prevent any type of electrical fire, special care has to be taken to ensure only the sensing element of this design to be exposed to the workspace environment and rest of the component and circuitry has to be isolated via an airtight container.

We can also use a relay in the design to switch off the entire circuit after gas/smoke detection but that will limit functionality and not indicate whether inflammable gas is still present after turn off, making it difficult for repair crews or maintenance staff to figure out presence/cause of leak and also whether or not it is safe to attempt repairs as there will be no indication of level of gas after the circuit has been switched off.

Conclusion

This paper has satisfactorily fulfilled all the basic requirement for solution of the problem stated i.e. - to reduce risk of fire due to leakage of inflammable gases in an industrial or commercial setting. This automatic system remove necessity for human intervention and is extremely reliable and accurate. This system also shows that overall workplace safety can be improved using similar systems and reduce risk and improve efficiency and productivity.

Also due to the 'Always On' feature of this system it becomes extremely safe and easy for repair and maintenance crews to assess and work on required changes and repairs to return the system to normal operation after any fault has occurred.

References

1. A complete reference of Micro Controllers, "Natwar Singh".
2. The 8051 microcontroller and embedded systems "Muhammad Ali Mazidi"
3. 8051 Microcontroller: An Applications Based Introduction by "David Calcutt, Frederick Cowan, and Hassan Parchizadeh"
4. C and the 8051 by Tom Schultz.
5. Der Keil C51-Compiler by "Michael Baldischweiler"