NYU TANDON SCHOOL OF ENGINEERING

SPRING 2016

# ME-5913-Mechatronics

## Integrated Term Project
## Final Report

## Gas Detecting Safety Hat (GDSH)

Shweta M. Vaviya
Angad A. Boralkar
Ahmad I. Alaqeely

05/02/2016

## Abstract

The objective of this project is to build a smart safety helmet that can detect toxic gases and notify field workers of the presence of toxic gas in work vicinity so they can immediately clear out the work area. This is accomplished by integrating a safety hat with a toxic gas alarm system that is composed of a microcontroller, a buzzer (sound alarm), a light emitting diode (LED) and a RF transmitter. Another stand-alone alarm system, which should theoretically be placed in facilities control room, is wirelessly synchronized with the alarm system mounted on the hat to notify the control room operator, who can subsequently announce the gas release and request all workers to vacate the operation area.

# Table of Contents


INTRODUCTION ..... 4
THEORY ..... 4
MATHEMATICAL BACKGROUND ..... 5
MECHANICAL DESIGN ..... 7
ELECTRONIC CIRCUITS ..... 8
BOM ..... 10
PROTOTYPE COST ..... 10
COST ANALYSIS FOR MASS PRODUCTION ..... 11
ADVANTAGES AND AREAS OF IMPROVEMENT ..... 12
TECHNICAL BASE FOR SELECTING SENSOR, ACTUATOR, IC, MECHANISMS ..... 12
THOROUGHLY COMMENTED ARDUINO IDE CODE ..... 13
REFERENCES ..... 19
APPENDIX ..... 20

## Introduction

Gases and vapors, such as $H_2S$, $CH_4$, released from oil, gas and petrochemical processing in addition to mining activities can, under many circumstances, have harmful effects on workers exposed to them by inhalation, being absorbed through the skin, or swallowed. People exposed to harmful substances may develop terminal illnesses such as cancer many years after the first exposure. Many toxic substances are dangerous to health in low 'ppm' (parts per million) or even ppb (part per billion) concentrations.

Given that toxic gases are often invisible and/or odorless, they are significantly dangerous. Concentration of the order 10,000 ppm is equivalent to 1% volume of any space, it can be seen that an extremely low concentration of some toxic gases can present a hazard to health.

According to a study conducted by the Organization for Economic Co-operation and Development (OECD) which analysis the $H_2S$ incidents in industry, one of the key factors leading to gas release (H2S only) related fatalities is the lack of The lack of suitable gas alarms and personal protective equipment led, in a number of cases, to fatal H2S exposure.

Although stand-alone gas detectors are already available in the market and also in each facility shop, people tend to be sloppy and often forget to take those devices with them when entering a processing facility. However, the one thing they always wear inside plants is their safety helmets, because if they don't, they wouldn't be allowed to enter the plant premises. So the only reasonable way to overcome this sloppiness is to eliminate the human error factor. Therefore, in this project, it is proposed to add the gas detector to be an integral part of the safety hat in a way that will alarm people when there is a gas release in the area. The idea is to have a microcontroller mounted on a hat along with a gas sensor that will detect a minimum threshold gas concentration above which workers could be harmed, and a signal will be sent to the actuator, which can be a buzzer and/or LED, to warn the workers and they should immediately clear the area.

## Theory

The alarm system is made of two subsystems, the self-calibrating *main* alarm system that is installed on the hat (will be referred to as *main* hereinafter), and the *auxiliary* alarm system that is placed in the control room (will be referred to as *auxiliary* hereinafter). The *main* system uses an Arduino microcontroller that is powered externally by a 9 VDC battery. The microcontroller is connected to an external breadboard on which a gas sensor (MQ-2), a light emitting diode (LED), a Piezoelectric Speaker and an RF 433 transmitter are mounted. When the gas sensor detects an LPG gas above the threshold set in the code, a high signal is generated by the microcontroller to light up the LED and power up the Piezoelectric Speaker alarm. At the same time, the transmitter sends a wireless signal to the receiver, which is part of the auxiliary system.

The auxiliary system is also built using an arudino microcontroller that is powered by an external 9 VDC battery and it consists of an RF 433 receiver, a light emitting diode (LED) and a Piezoelectric Speaker. When the receiver gets the signal from the transmitter, a signal is generated and it passes through the circuit to blink the LED and fire up the Piezoelectric Speaker. Below is a simple flow chart that explains how the system should work in theory.

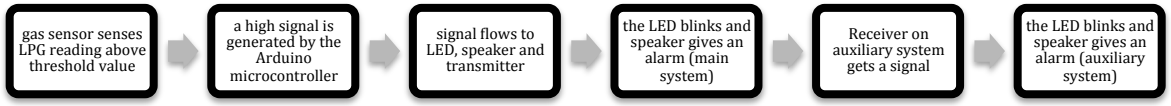



Figure 1: How the alarm system works

## Mathematical Background

The alarm system is basically an integration of several components, each of which has its own mathematical model that describes the way it functions.

MQ-2 Gas Sensor

The sensor gas concentration measurements and calibration is based on the following sensitivity characteristic chart.

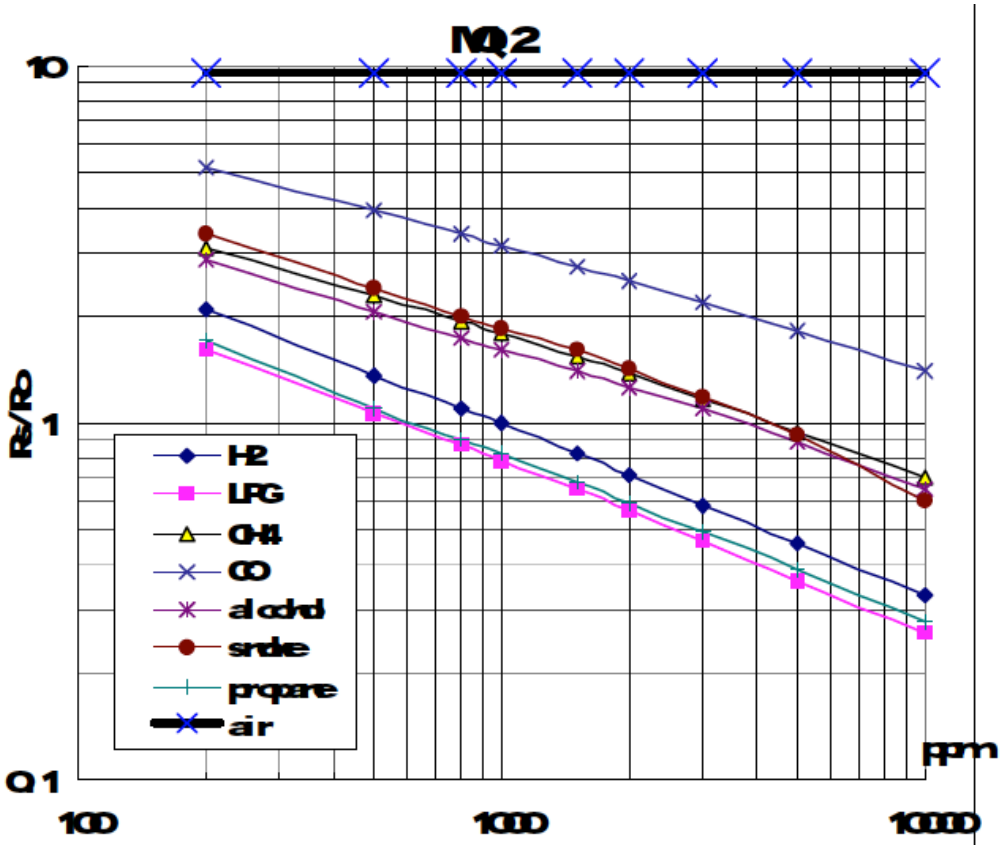



Figure 2: Sensitivity Characteristics of MQ-2

To calculate the gas concentration, the LPG curve (pink) is linearized by taking the log of two end points and finding the slope. In the above figure, the y-axis represents a fraction of sensor resistance at various concentrations of gases ($R_s$), with respect to the sensor resistance at 1000ppm of H2 in the clean air ($R_o$). The x-axis represents the measured gas concentration in ppm. Slope is calculated as

$$m = (\log y_2 - \log y_1)/(\log x_2 - \log x_1)$$

Then, this slope value is used to generate a straight-line equation in the following format

$$\log y = m \log x + \log b$$

Where b is the offset value in y direction. After rearrangement, the equation becomes

$$x = \left(\frac{y}{b}\right)^{\frac{1}{m}}$$

Piezoelectric Speaker

The piezoelectric speaker receives electrical power from the microcontroller and generates mechanical energy in the form of sound. The electrical power is the result of the voltage supplied by the microcontroller and the current in between and is defined as follows

$$P = VI$$

Then, this electrical power is transferred into Mechanical power in the form of sound to the surrounding environment. This mechanical power is defined as

$$P = pQ$$

Where $p$ is the air pressure and $Q$ is the flow rate.

Light Emitting Diode

According to a study titled "Design Methodology for Light-Emitting Diode Systems by Considering an Electro-thermal Model" conducted by a committee consisted of several IEEE members, LED electrical behavior is effected by high ohmic losses, normally represented as parasitic resistances. The voltage behavior of an LED is described by Shockley equation, including the parasitic series resistance (Rs) effect, as shown in the following equation

$$V_f = \frac{n_i k_b T_j}{q_e} \ln\left(\frac{I_f}{I_s}\right) + \frac{E_g T_j}{q_e} + R_s I_f$$

Where $V_f$ is the forward voltage, $I_f$ is the forward current, $I_s$ is the reverse saturation current, $q_e$ is the electron charge, $n_i$ is the ideality factor, $k_b$ is the Boltzmann constant, $T_j$ is the junction temperature, and $E_g$ is the band gap energy. The LED voltage is expressed as the sum of a linear and logarithmic function of the current.

Luminous flux is a parameter strongly dependent on two variables, forward current and junction temperature. The mathematical description of luminous flux behavior as function on the junction temperature is

$$F(T_i) = c_0 + c_1 T_j$$

Where F represents the luminous flux, $c_0$ is the linear coefficient of luminous flux dependency with junction temperature and $c_1$ is the angular coefficient obtained by curve regression.

RF 433 Transmitter and Receiver

Radio Frequency (RF) signal is a controlled variation of electrical energy over time. This variation is manifested in intensity fluctuation between a certain minimum and maximum, which can be described using a sinusoidal function, characterized by its frequency and amplitude.

The wavelength is the distance a radio wave will travel during one cycle. The formula that relates the wavelength to the frequency is the following:

$$\lambda = \frac{c}{f}$$

Where $\lambda$ is the wavelength, $c$ is the speed of light and $f$ is the frequency.

The power of a constant low-frequency signal is expressed in terms of amplitude (voltage) and current. In a DC circuit, a source of V volts, delivering I Amperes, produces P Watts according to the formula:

$$P = VI$$

The logarithmic expression of the ratio between the power, voltage, or current of two signals is known as the *decibel* (dB). The power and voltage ration expressed in *decibels* are

$$P_{dB} = 10\log(\frac{P_2}{P_1})$$

$$V_{dB} = 20\log(\frac{V_2}{V_1})$$

## Mechanical Designs

This system is an integration of readily available electrochemical and electromechanical devices forming an electrical circuit by use of appropriate passive and communication elements. More details on the design, parts used and circuit are provided in the subsequent sections.

## Electronic Circuits

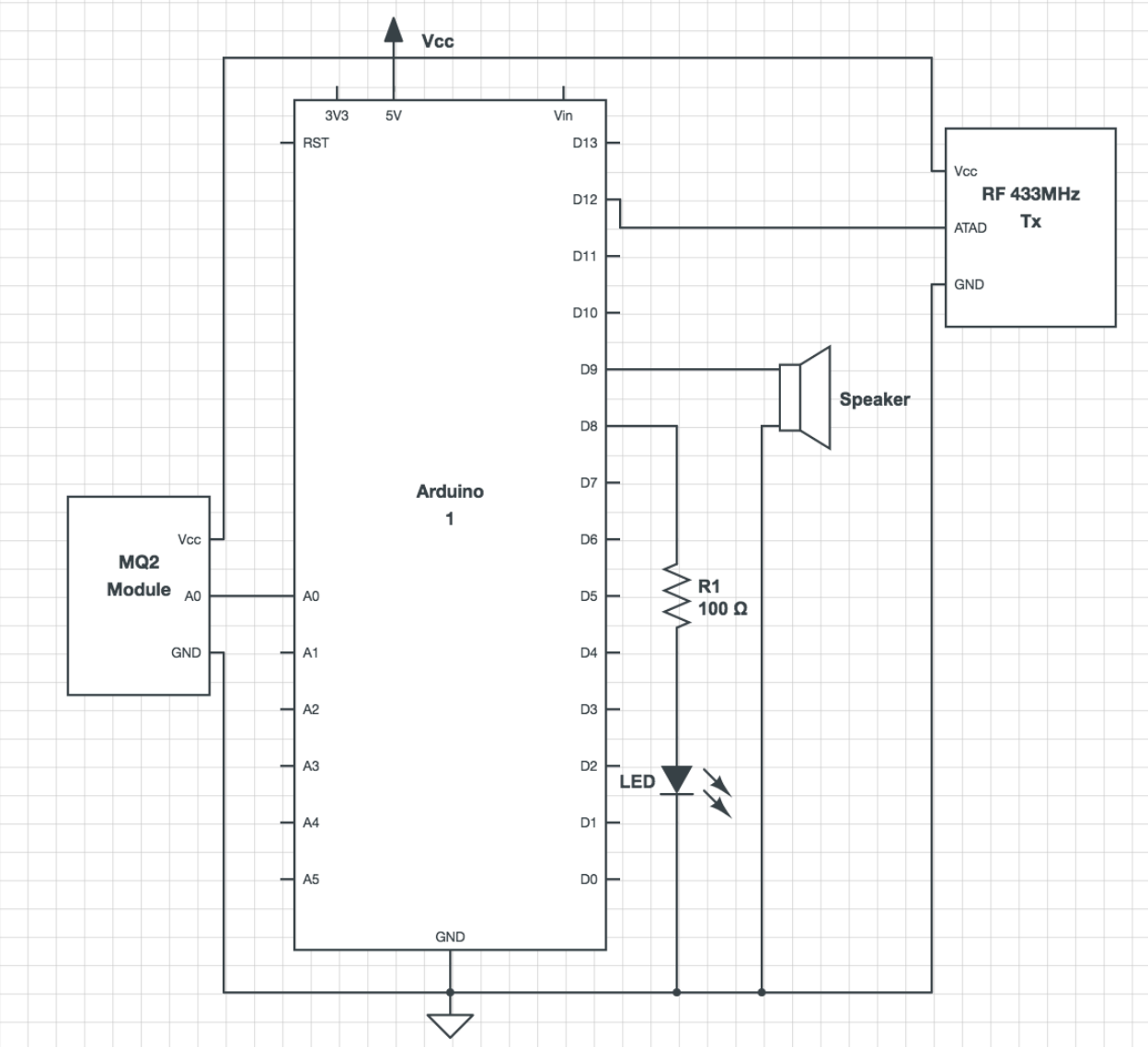



Figure 3: Main System Circuit Schematic

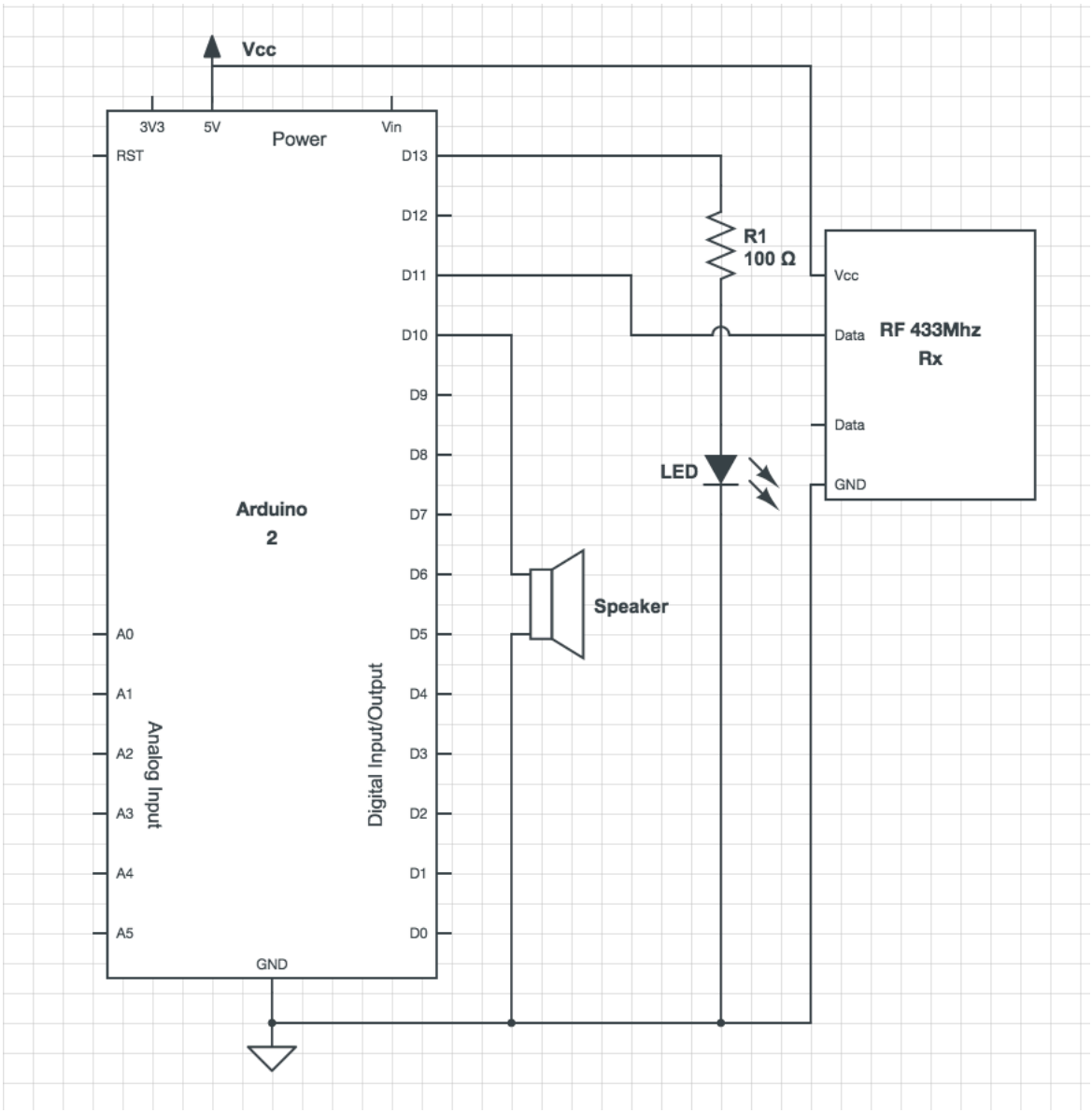



Figure 4: Auxiliary System Circuit Schematic

## Bill-of-Materials

Below is a list of all materials used in this project

| No. | Item Description | Quantity |
|---|---|---|
| 1 | Arduino Microcontroller | 2 |
| 2 | MQ-2 Sensor | 1 |
| 3 | Piezoelectric Speaker | 2 |
| 4 | Light Emitting Diode | 2 |
| 5 | External breadboard | 2 |
| 6 | USB Communication Cable | 2 |
| 7 | Battery; 9 VDC | 2 |
| 8 | 433 MHz RF transmitter | 1 |
| 9 | 433 MHz RF receiver | 1 |
| 10 | Resistor; 100 ohm | 1 |
| 11 | Jumper wires | 10 |
| 12 | Safety hat | 5 |

## Prototype Cost

The prototype total cost is $98.83. Below is a cost breakdown for all components of the prototype.

| No. | Item Description | Quantity | Cost per Unit ($) | Total Cost ($) |
|---|---|---|---|---|
| 1 | Arduino Microcontroller and Shield | 2 | 21.54 | 43.08 |
| 2 | MQ-2 Sensor | 1 | 7.63 | 7.63 |
| 3 | Piezoelectric Speaker | 2 | 1.95 | 3.90 |
| 4 | Light Emitting Diode | 2 | NA | 4.1 |
| 5 | External breadboard | 2 | NA | 6.99 |
| 6 | USB Communication Cable | 2 | 2.09 | 4.18 |
| 7 | Battery; 9 VDC | 2 | 2.66 | 5.32 |
| 8 | 433 MHz RF transmitter | 1 | 10.65 | 10.65 |
| 9 | 433 MHz RF receiver | 1 | NA (comes with transmitter) | NA |
| 10 | Resistor; 100 ohm | 1 | NA | 3.14 |
| 11 | Jumper wires | 10 | NA | 4.84 |
| 12 | Safety hat | 1 | 5 | 5 |
| | | | **Total** | 98.83 |

## Cost Analysis for Mass Production

To start a mass production, one must make a commitment to suppliers for the initial order, which is typically the largest component of start-up cost. The minimum order quantity for any new product, assuming acceptance by suppliers and end users, could easily be 50,000 to 100,000 units. The total cost for building this system could be further reduced through a wholesale since some components per unit are supplied in quantities larger than the amount needed for a single system such as jumper wires, resistor,..etc.
To accurately determine the mass production cost, it is essential to first determine the exact quantities needed of each component. The below analysis is a breakdown of system components and the quantities needed to develop 50,000 systems.

| No. | Item Description | Quantity Needed Per System | Quantity Supplied by Vendor Per Package | Quantities To Be Ordered for Mass Production |
|---|---|---|---|---|
| 1 | Arduino Microcontroller & Shield | 2 | 1 | 100,000 |
| 2 | MQ-2 Sensor | 1 | 1 | 50,000 |
| 3 | Piezoelectric Speaker | 2 | 1 | 100,000 |
| 4 | Light Emitting Diode | 2 | 500 | 200 |
| 5 | External breadboard | 2 | 6 | 16,667 |
| 6 | USB Communication Cable | 1 | 1 | 50,000 |
| 7 | Battery; 9 VDC | 1 | 1 | 50,000 |
| 8 | 433 MHz RF transmitter & receiver | 1 | 1 | 50,000 |
| 9 | Resistor; 100 ohm | 1 | 600 | 84 |
| 10 | Jumper wires | 10 | 140 | 3572 |
| 11 | Safety hat | 1 | 1 | 50,000 |

Next, the cost for each component needed to produce 50,000 units is calculated and the total mass production cost is generated as shown in the following table

| No. | Item Description | Quantity Needed Per System | Quantities To Be Ordered for Mass Production | Cost for Mass Production ($) |
|---|---|---|---|---|
| 1 | Arduino Microcontroller & Shield | 2 | 100,000 | 2,154,000 |
| 2 | MQ-2 Sensor | 1 | 50,000 | 381,500 |
| 3 | Piezoelectric Speaker | 2 | 100,000 | 195,000 |
| 4 | Light Emitting Diode | 2 | 200 | 820 |
| 5 | External breadboard | 2 | 16,667 | 116,502 |
| 6 | USB Communication Cable | 2 | 100,000 | 209,000 |
| 7 | Battery; 9 VDC | 2 | 100,000 | 266,000 |
| 8 | 433 MHz RF transmitter & receiver | 1 | 50,000 | 532,500 |
| 9 | Resistor; 100 ohm | 1 | 84 | 263.76 |
| 10 | Jumper wires | 10 | 25,000 | 17,288 |
| 11 | Safety hat | 1 | 50,000 | 250,000 |
| | | | **Total** | 4,122,873,76 |

Finally, applying discount factor of 10% that is usually associated with wholesales, a cost reduction of $412,287.376 is achieved, leading to a total cost of $3,710,586.384.

Comparing this cost to the cost of producing 50,000 units individually without mass production and with $98.83 per unit, which sums to $4,941,500, a significant cost saving of 33.17% is realized.

## Advantages and Areas of Improvement

One of the advantages of this system is that it will significantly reduce the risk factor by alarming the workers well in advance, so they have enough time to evacuate the place. This will also eliminate the need to have two separate devices and replace them with a single, light and compacted system integrated into an essential gear that workers never forget to wear inside plants and concurrently, eliminating the human error factor from the situation. In terms of cost, a considerable cost reduction is realized when looking at a stand-alone gas sensor, which could go as high as $500 in cost, compared to GDSH that costs $98.83. This GDSH has a wide area of application from oil, gas & petrochemical facilities all the way to mining fields and manholes. Lastly, the system is self-calibrating and does not need human intervention.

One limitation of GDSH, though, is that the gas sensor selected for this project is more of a testing type sensor used for educational purposes and may not be very accurate in field applications. Moreover, this sensor is not designed for long term and extensive applications and might need to be replaced with a more accurate sensor, which will lead to a cost increase. Another limitation is the way this system is mounted on the hat (which was not a substantial consideration in the scope of this project) that subjects the system to possible detaching from the hat if used in the field and potentially affecting the accuracy of the gas sensor if exposed to dust, rain or heat. This limitation could be resolved if the system was re-designed implementing micro or Nano technology design concepts resulting in a better compacted and protected system. Finally, the wireless subsystem requires that both transmitter & receiver be at a close range of each other. A more advanced transmitter and receiver could be incorporated to enhance distance coverage and overall reliability of the wireless subsystem.

## Technical Base for selecting sensor, actuator, IC, mechanisms

MQ2 Gas Sensor:

This sensor was selected for two main reasons. One, it is a fairly cheap sensor with enough accuracy required for making a prototype of the project. Second, it can detect propane gas which is the least harmful gas that could be used for demonstration purposes. There are other gas sensors belonging to the same family like the MQ3, MQ4 etc. which can detect gases like alcohol, methane etc. respectively. The drawback of using these sensors instead of the MQ2 is that gases like methane and alcohol are more difficult to procure in pure form and they are more hazardous to health than propane.

433MHz RF Transmitter and Receiver:

This RF module is used transmit signal wirelessly to the Arduino. The major reason behind choosing this module is that it is significantly cheaper than its counterparts such as the Xbee wireless transceiver. Unlike other modules, which use sophisticated protocol for wireless communication, this module works on a fairly simple wireless protocol and it is easy to install. No predefined configuration or initial setup is required to use this RF module. The 433MHz

frequency band is used as it is an uncommon frequency and little to none of the common radio devices use this band for transmission. This factor helps in reducing the noise/interference from the surrounding in the circuit while transmitting signals.

Arduino UNO Microcontroller:

The Arduino UNO microcontroller is used in this project because this board is powerful enough to deliver satisfactory results. The microcontroller has its own software (Arduino IDE) for coding which is very easy and convenient to use. Additionally, this microcontroller is backed by a many useful libraries and resources which can be used for free due to Arduino popularity all over the world. Another reason this controller was chosen is that it is of a perfect size and easier to mount on the safety hat used for demonstration. Other microcontrollers like the BS2 do not have as strong support structure as the Arduino and also is bigger in size.

Prototyping Shield for the Arduino:

A separate shield was used for the transmitter circuit. This shield is called the Prototyping shield and it fits right on top of an Arduino UNO microcontroller. It is a retro fit electronic board that has the exact same form factor as of the Arduino UNO. The use of this shield allows for a cleaner working environment for circuits without sacrificing the use of all I/O pins of the original Arduino board. This shield gives a cleaner look to the prototype with minimal jumper wires used in the circuit.

## Thoroughly Commented Arduino IDE Code

Main System

```
/*****************GAS DETECTING SAFETY HELMET  ********************/

#include<VirtualWire.h>                          //including library VirtualWire for RF transmission

/********************** Defining Constants  *********************************/
#define      MQ_PIN                  (0)          //analog Pin used to get input from MQ-2
#define      RL_VALUE                (5)          //load resistance on the board, in kilo ohms
#define      RO_CLEAN_AIR_FACTOR  (9.83)          //RO_CLEAR_AIR_FACTOR=(Sensor resistance in clean air)/RO, which is derived from the chart in datasheet

#define      CALIBARAION_SAMPLE_TIMES    (50)     //number of samples taken in the calibration phase

#define      CALIBRATION_SAMPLE_INTERVAL  (500)   //time interval (in millisecond) between each samples in the calibration phase
#define      READ_SAMPLE_INTERVAL         (50)    //number of samples taken in normal operation
```

```
#define        READ_SAMPLE_TIMES          (5)     //time interval(in millisecond) between
                                                    each samples in normal operation

#define        GAS_LPG                    (0)          //LPG Gas ID number

/****************************** Global ******************************/

float          LPGCurve[3]  =  {2.3,0.21,-0.47};   //two points are taken from the curve. with
                                                    these two points, a line is formed which is
                                                    "approximately equivalent" to the original
                                                    curve. data format:{ x, y, slope}; point1:
                                                    (lg200, 0.21), point2: (lg10000, -0.59)

float          Ro          =  10;                  //Ro is sensor resistance at 1000ppm of H2
                                                    in the clean air and is initialized to 10 kilo
                                                    ohms

/************************** PIN Definitions **************************/
int pin8 = 8;                                      //pin for LED on main system
int pin9 = 9;                                      //pin for Buzzer on main system
char msg[1] = {'0'};

void setup()
{
 pinMode(pin8, OUTPUT);                            //defining pins as output pins (main)
 pinMode(pin9, OUTPUT);

 Serial.begin(9600);                               //UART setup, baud-rate = 9600bps
 vw_set_tx_pin(12);                                //set pin 12 as transmit pin
 vw_setup(2000);                                   //speed of data transfer bps

 Serial.print("Calibrating...\n");
 Ro = MQCalibration(MQ_PIN);                       //calibrating the sensor. Please make sure
                                                    the sensor is in clean air when you
                                                    perform the calibration
 Serial.print("Calibration is done...\n");
 Serial.print("Ro=");
 Serial.print(Ro);                                 //value of Ro in k-ohms after calibrating

 Serial.print("kohm");
 Serial.print("\n");
}

void loop()
{

  Serial.print("LPG:");
ppm(GAS_LPG);                                      //calling the function ppm() to get the calculated
```

```
                                        //ppm of LPG
  Serial.print( "ppm" );
  Serial.print("    ");
  Serial.print("\n");
  delay(100);

  vw_send((uint8_t *)msg, 1);           //send char msg to receiver
  vw_wait_tx();                         //wait until the whole message is gone
  delay(100);
}

/************************MQResistanceCalculation*************************

Input:   raw_adc - raw value read from adc, which represents the voltage
Output:  the calculated sensor resistance
Remarks: The sensor and the load resistor forms a voltage divider. Given the voltage across
the load resistor and its resistance, the resistance of the sensor could be derived.
************************************************************************
float MQResistanceCalculation(unsigned int raw_adc)
{
 return ( ((float)RL_VALUE*(1023-raw_adc)/raw_adc));
}

/*****************************MQCalibration****************************
Input:   mq_pin - analog channel
Output:  Ro of the sensor
Remarks: This function assumes that the sensor is in clean air. It use
MQResistanceCalculation to calculates the sensor resistance in clean air and then divides it
with RO_CLEAN_AIR_FACTOR. RO_CLEAN_AIR_FACTOR is about 10, which
differs slightly between different sensors.
************************************************************************/
float MQCalibration(int mq_pin)
{
 int i;
 float val=0;

 for (i=0;i<CALIBARAION_SAMPLE_TIMES;i++) {          //take multiple samples
  val += MQResistanceCalculation(analogRead(mq_pin));
  delay(CALIBRATION_SAMPLE_INTERVAL);
 }
 val = val/CALIBARAION_SAMPLE_TIMES;                 //calculate the average value

 val = val/RO_CLEAN_AIR_FACTOR;                      //divided by RO_CLEAN_AIR_FACTOR
                                                     yields the Ro according to the chart in the
                                                     datasheet

 return val;
}
```

```
/****************************MQRead *********************************
Input:   mq_pin - analog channel
Output:  Rs of the sensor
Remarks: This function use MQResistanceCalculation to calculate the sensor resistance
(Rs).The Rs changes as the sensor is in the different concentration of the target gas. The
sample   times and the time interval between samples could be configured by changing the
definition of the macros.
************************************************************************/
float MQRead(int mq_pin)
{
 int i;
 float rs=0;

 for (i=0;i<READ_SAMPLE_TIMES;i++) {                    //take multiple samples
  rs += MQResistanceCalculation(analogRead(mq_pin));
  delay(READ_SAMPLE_INTERVAL);
 }

 rs = rs/READ_SAMPLE_TIMES;                             //calculate the average value

 return rs;
}

/*************************MQGetGasPercentage ***************************
Input:   rs_ro_ratio - Rs divided by Ro
        gas_id      - target gas type
Output:  ppm of the target gas
Remarks: This function passes different curves to the MQGetPercentage function which
calculates the ppm (parts per million) of the target gas.
************************************************************************/
unsigned int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
{
 if ( gas_id == GAS_LPG ) {
   return MQGetPercentage(rs_ro_ratio,LPGCurve);
 }
 return 0;
}

/****************************MQGetPercentage ***************************
Input:   rs_ro_ratio - Rs divided by Ro
        pcurve      - pointer to the curve of the target gas
Output:  ppm of the target gas
Remarks: By using the slope and a point of the line. The x(logarithmic value of ppm) of the
line could be derived if y(rs_ro_ratio) is provided. As it is a logarithmic coordinate, power
of 10 is used to convert the result to non-logarithmic value.
************************************************************************/
unsigned int  MQGetPercentage(float rs_ro_ratio,float *pcurve)
{
```

```
  return (pow(10,( ((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0])));
}


/*************Calculating ppm and defining threshold level conditions   *************/

void ppm(int gas_id)
{

  unsigned int ppm;

  if ( gas_id == GAS_LPG )
    {
         delay(10);
        ppm=MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG);
        if(ppm>4000)                                  //setting threshold for LPG gas as 4000ppm
        {
          ledBlink();                                 //blink LED for LPG ppm above 4000
          msg[0]='1';                                 //send "1" to receiver
          tone(9,200,20);                             //beep the buzzer LPG ppm above 4000
        }
        else
        {
          msg[0]='0';                                  //send "0" to receiver
          digitalWrite(pin8, LOW);                     //deactivating pins for gas values below the
                                                        threshold
          digitalWrite(pin9, LOW);
        }

       if(ppm>10000)
          Serial.print(" Above Range ");               //for LPG ppm values above 10000ppm
       else
          Serial.print(ppm);                           //LPG ppm on Serial monitor
    }

}


/************************Blink LED on main system ************************/
void ledBlink()
{
  digitalWrite(pin8, HIGH);                            //send high signal to LED
  delay(10);
  digitalWrite(pin8, LOW);                             //send low signal to LED
  delay(10);
}
```

Auxiliary System

```
/*-------------------------------------------------------------------------*/
/********************** Auxiliary System ********************************/
/* This code is used on the receiver end. The receiver end has a separate Arduino connected
to it and it gives an alarm in the form of LED and Buzzer when the transmitter sends a
signal to it. This receiver system is shown as a demonstration of control room system. */

#include <VirtualWire.h>                  //including library VirtualWire for RF transmission

/************************** Arduino Pin definitions ***********************/

const int led_pin = 13;
const int buzzer_pin = 10;
const int transmit_pin = 12;
const int receive_pin = 11;

/************************** Initial Setup of Arduino **********************/

void setup()
{

   Serial.begin(9600);
   Serial.println("setup");

   vw_set_tx_pin(transmit_pin);                  //assign a pin number to the transmitter
   vw_set_rx_pin(receive_pin);                   //assign a pin number to the receiver

   vw_setup(2000);                               //speed of transmission in bits per sec
   vw_rx_start();                                //start the receiver. Program running.

   pinMode(led_pin, OUTPUT);
   pinMode(buzzer_pin,OUTPUT);
}

void loop()
{
   uint8_t buf[VW_MAX_MESSAGE_LEN];
   uint8_t buflen = VW_MAX_MESSAGE_LEN;

   if (vw_get_message(buf, &buflen))                    // Non-blocking
   {
     int i;

     for (i = 0; i < buflen; i++)
       {
         if (buf[i]==49)
           {
```

```
            digitalWrite(led_pin, HIGH);       //LED high when a signal is received from the
                                                 transmitter
            delay(10);
            digitalWrite(led_pin, LOW);
            delay(10);
            tone(buzzer_pin,200,20);             //beep the Buzzer when a signal is received from
                                                the transmitter
          }
        else
          {
           digitalWrite(led_pin, LOW);           //deactivating pins
           digitalWrite(buzzer_pin,LOW);
          }
      }
   }
}
```

**References**


"Toxic Risk" crowcon. Web. 02 Apr. 2016.

Danielsson, Fendler, Hailwood, Shrives. Analysis of H2S – incidents in geothermal and other industries: Preliminary analysis of data. 04 Sep. 2009.

"Electrochemical Gas Sensors" Wikipedia. Web. Jun.2008.

V. Bender, O. Iaronka, W.Vizzotto, M.Costa, R. Prado, T. Marchesan, "Design Methodology for Light-Emitting Diode Systems by Considering an Electrothermal Model". 11 Nov. 2013

"Basic Concepts- RF Basics" wireless.ictp.it.

"Manufacturing Cost Estimates for Inventions" octs.com

## Appendix A

Safety

The Arduino has a resettable polyfuse that protects computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Appendix B

System Images

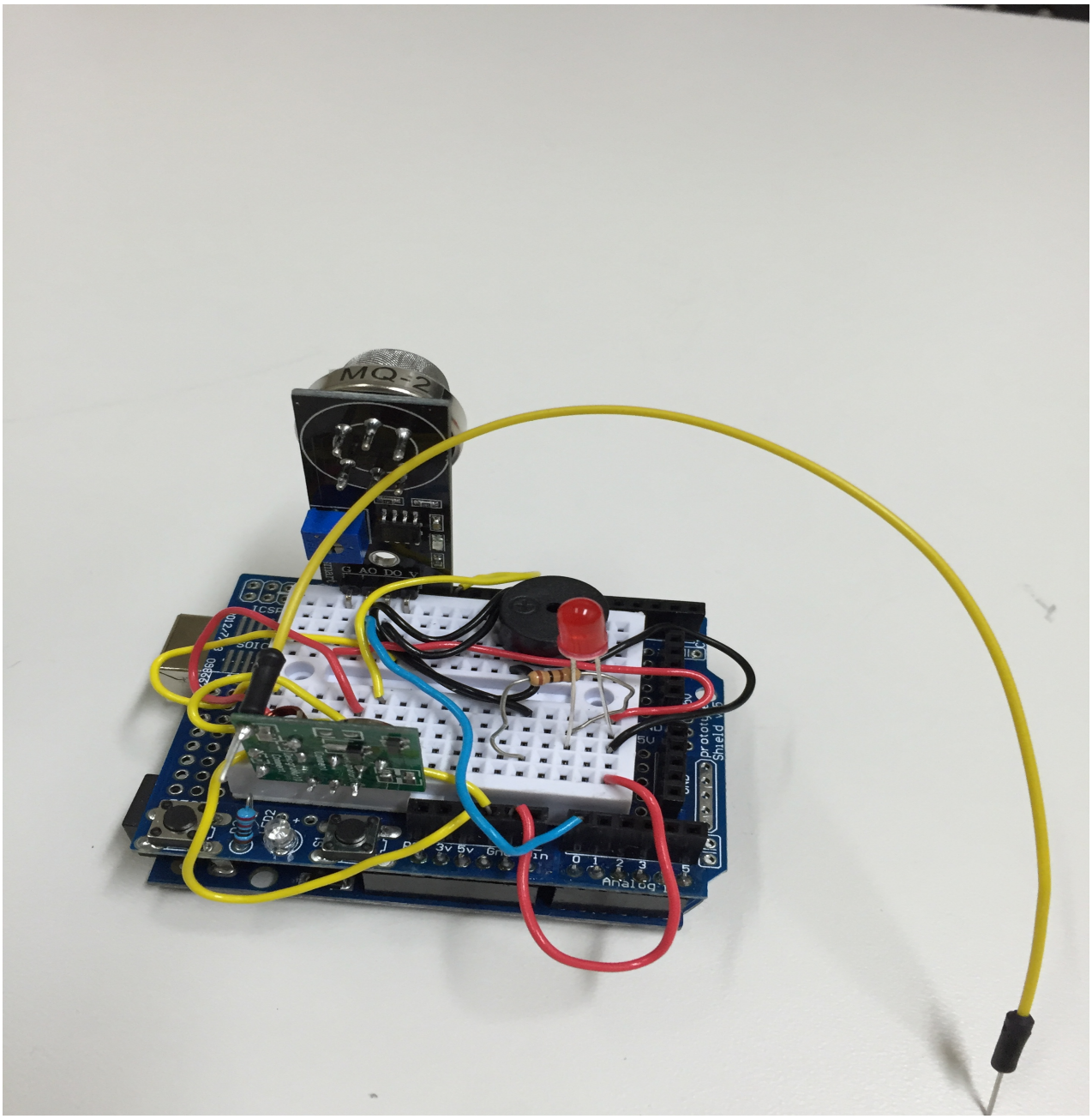



Figure 5: Main system before mounting on the hat

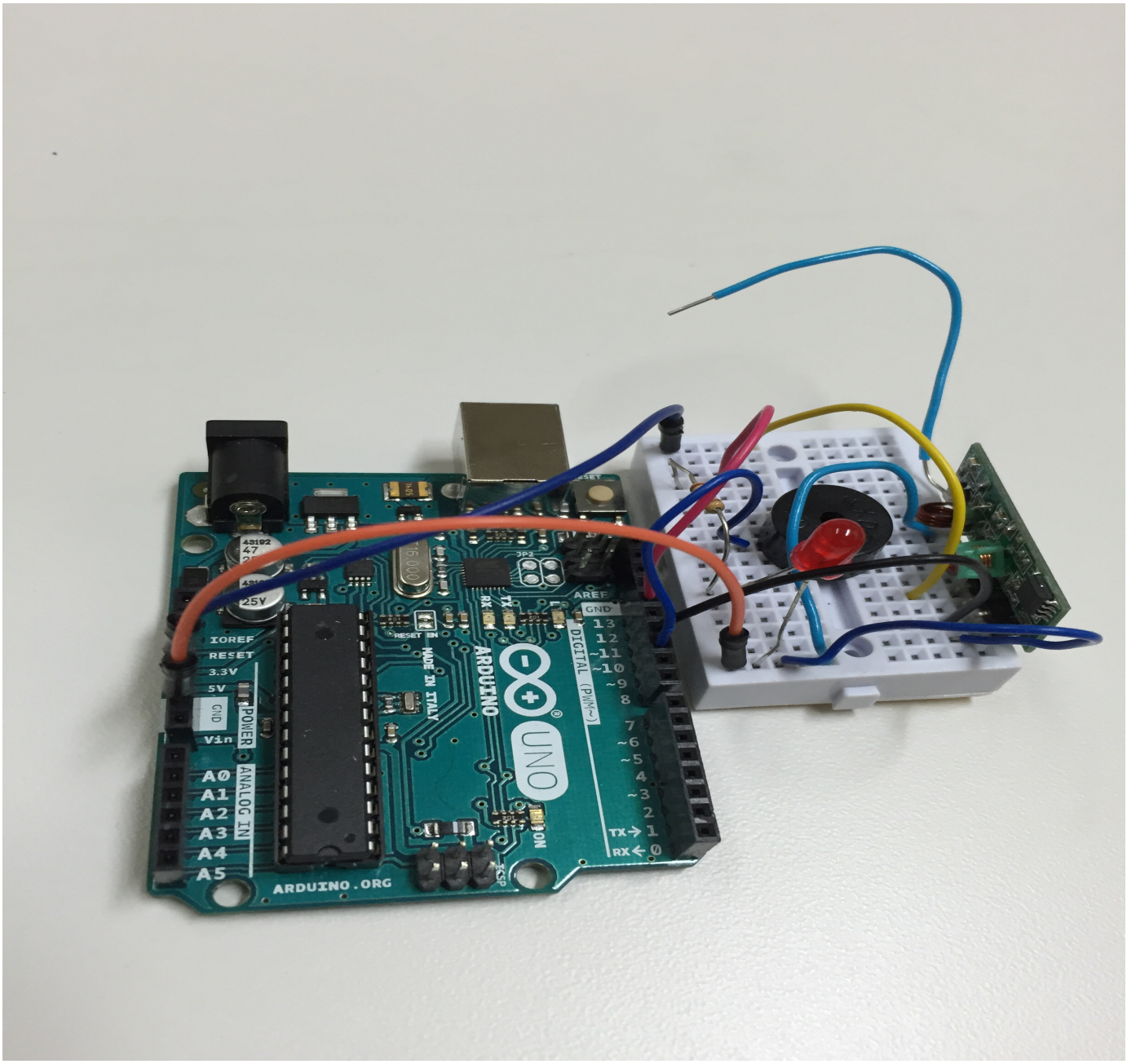



Figure 6: Auxiliary System

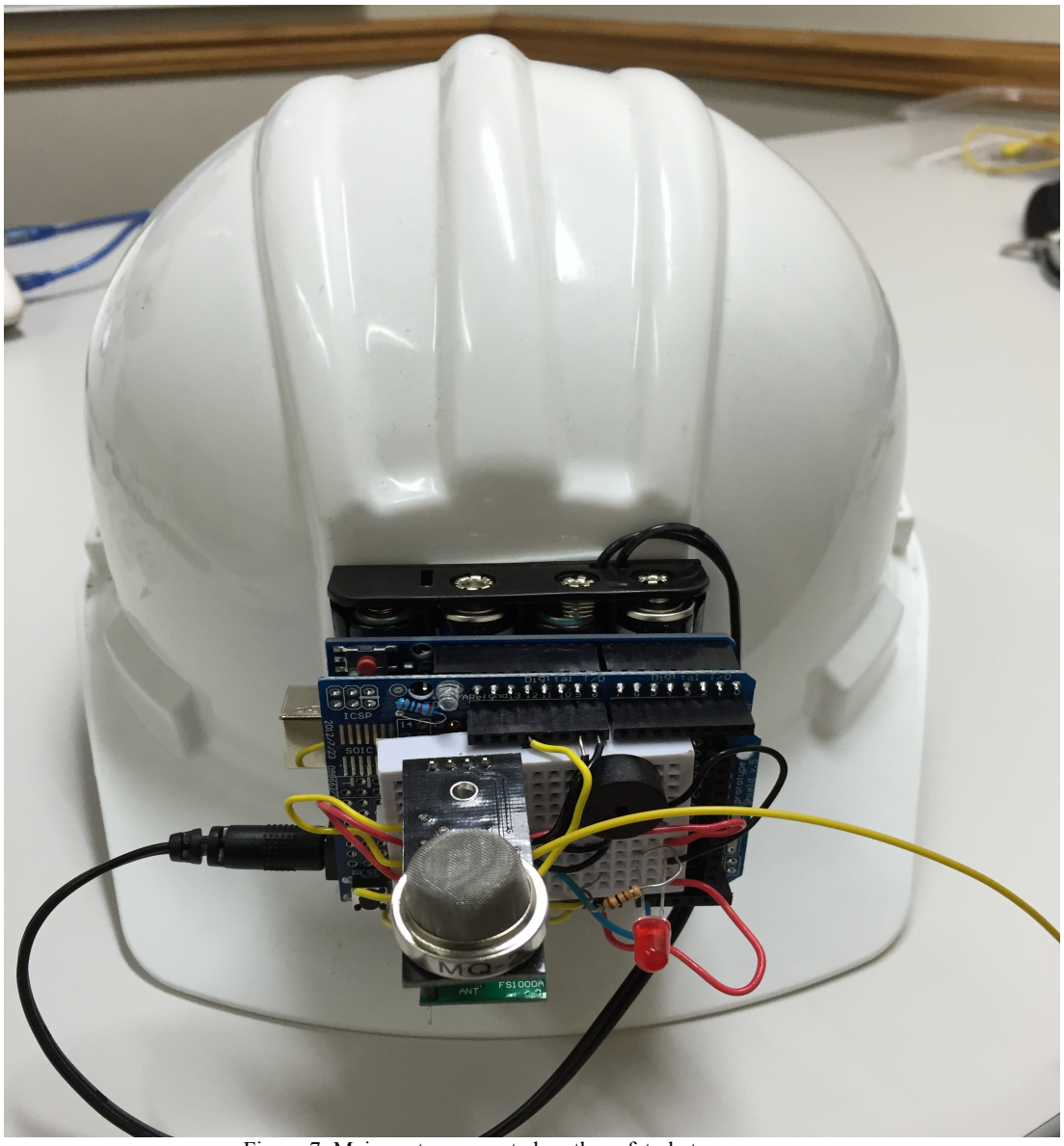


Figure 7: Main system mounted on the safety hat

## Appendix C

# TECHNICAL DATA MQ-2 GAS SENSOR

## FEATURES

Wide detecting scope — Fast response and High sensitivity
Stable and long life — Simple drive circuit

## APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

## SPECIFICATIONS

A. Standard work condition

| Symbol | Parameter name | Technical condition | Remarks |
|---|---|---|---|
| Vc | Circuit voltage | 5V±0.1 | AC OR DC |
| $V_H$ | Heating voltage | 5V±0.1 | ACOR DC |
| $R_L$ | Load resistance | can adjust | |
| $R_H$ | Heater resistance | 33Ω ±5% | Room Tem |
| $P_H$ | Heating consumption | less than 800mw | |

B. Environment condition

| Symbol | Parameter name | Technical condition | Remarks |
|---|---|---|---|
| Tao | Using Tem | -20℃-50℃ | |
| Tas | Storage Tem | -20℃-70℃ | |
| $R_H$ | Related humidity | less than 95%Rh | |
| $O_2$ | Oxygen concentration | 21%(standard condition)Oxygen concentration can affect sensitivity | minimum value is over 2% |

C. Sensitivity characteristic

| Symbol | Parameter name | Technical parameter | Remarks |
|---|---|---|---|
| Rs | Sensing Resistance | 3KΩ-30KΩ (1000ppm iso-butane ) | Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm $H_2$ 100ppm-2000ppm Alcohol |
| α (3000/1000) isobutane | Concentration Slope rate | ≤0.6 | |
| Standard Detecting Condition | Temp: 20℃±2℃ Humidity: 65%±5% | Vc:5V±0.1 Vh: 5V±0.1 | |
| Preheat time | Over 24 hour | | |

D. Structure and configuration, basic measuring circuit

| | Parts | Materials |
|---|---|---|
| 1 | Gas sensing layer | $SnO_2$ |
| 2 | Electrode | Au |
| 3 | Electrode line | Pt |
| 4 | Heater coil | Ni-Cr alloy |
| 5 | Tubular ceramic | $Al_2O_3$ |
| 6 | Anti-explosion network | Stainless steel gauze (SUS316 100-mesh) |
| 7 | Clamp ring | Copper plating Ni |
| 8 | Resin base | Bakelite |
| 9 | Tube Pin | Copper plating Ni |

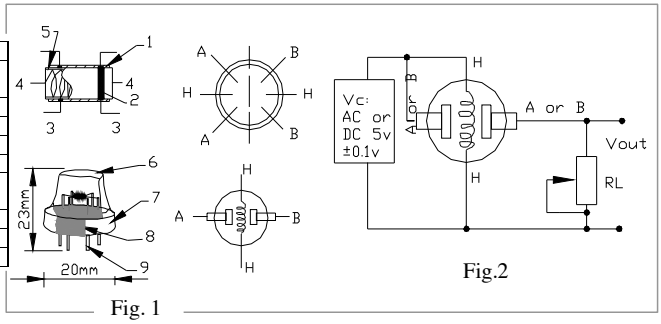
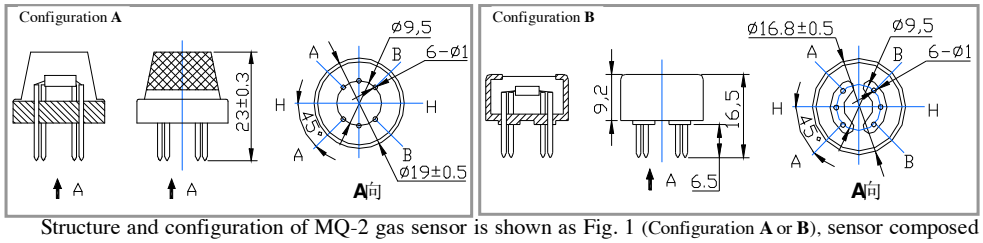



Fig. 1 Fig.2




Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration **A** or **B**), sensor composed by micro $AL_2O_3$ ceramic tube, Tin Dioxide ($SnO_2$) sensitive layer, measuring electrode and heater are fixed into a

**Transmitter :**

Working voltage: 3V - 12V fo max. power use 12V
Working current: max Less than 40mA max , and min 9mA
Resonance mode: (SAW)
Modulation mode: ASK
Working frequency: Eve 315MHz Or 433MHz
Transmission power: 25mW (315MHz at 12V)
Frequency error: +150kHz (max)
Velocity : less than 10Kbps

So this module will transmit up to 90m in open area .

**Receiver :**

Working voltage: 5.0VDC +0.5V
Working current:≤5.5mA max
Working method: OOK/ASK
Working frequency: 315MHz-433.92MHz
Bandwidth: 2MHz
Sensitivity: excel –100dBm (50Ω)
Transmitting velocity: <9.6Kbps (at 315MHz and -95dBm)

the use of an optional antenna will increase the effectiveness of your wireless communication. A simple wire will do the trick.

Arduino UNO R3 Specifications

| | |
|---|---|
| Microcontroller | ATmega328P |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P)<br>of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |